EIN ALLGEMEINES DEKOMPOSITIONSVERFAHREN FÜR LINEARE OPTIMIERUNGSPROBLEME

DISSERTATION

zur Erlangung des Grades

eines Doktors der Wirtschaftswissenschaft

der Rechts- und wirtschaftswissenschaftlichen Fakultät

der Universität des Saarlandes

(doctor rerum oeconomicarum)

vorgelegt von

Diplom-Kaufmann HERGEN HEINEMANN

aus Oldenburg (Oldb), 1971

EIN ALLGEMEINES DEKOMPOSITIONSVERFAHREN FÜR LINEARE OPTIMIERUNGSPROBLEME

DISSERTATION

zur Erlangung des Grades

eines Doktors der Wirtschaftswissenschaft

der Rechts- und wirtschaftswissenschaftlichen Fakultät

der Universität des Saarlandes

(doctor rerum oeconomicarum)

vorgelegt von

Diplom-Kaufmann HERGEN HEINEMANN

aus Oldenburg (Oldb), 1971

UNIV.
LEIDEN
BIBL.

Tag der Disputation : 3o. 6. 1971

Dekan : Prof. Dr. Geck
Erstberichterstatter : Prof. Dr. Hax
Zweitberichterstatter : Prof. Dr. Schneeweiß

Druck: aku- Fotodruck, 86 Bamberg, Färbergasse 8 R

Inhaltsverzeichnis

Seite

1. Vorbemerkungen 1

1.1. Allgemeines 1

1.1.1. Mögliche Vorteile eines allgemeinen Dekompositionsverfahrens für lineare Optimierungsprobleme 1

1.1.2. Bereits vorhandene Dekompositionsverfahren 2

1.1.2.1. Spezielle Verfahren 2

1.1.2.2. Generelle Verfahren 3

1.2. Allgemeine Anwendbarkeit des zu beschreibenden Verfahrens 5

2. Ein generelles Dekompositionsverfahren für lineare Optimierungsprobleme 6

2.1. Kurze Skizzierung des Verfahrens 6

2.2. Genaue Beschreibung des Verfahrens 7

2.2.1. Bildung von Teilproblemen 8

2.2.2. Lösen der Teilprobleme 12

2.2.3. Aufstellung eines Koordinierungsproblems 16

2.2.4. Lösen des Koordinierungsproblems (KOP) 20

2.2.5. Test auf Verbesserungsfähigkeit des KOP 27

2.3. Beweis für die Gleichwertigkeit der Optima des UP und des letzten KOP bei normaler Beendigung des Verfahrens und Endlichkeitsbeweis 32

Seite

2.3.1. Beweis für die Gleichwertigkeit der Optima 32

2.3.2. Endlichkeitsbeweis 37

2.4. Zweistufige Dekomposition 38

3. Erste praktische Tests des Algorithmus auf elektronischen Datenverarbeitungsanlagen 40

3.1. Das Testprogramm 40

3.2. Die Rechentests 40

3.2.1. Die Testbeispiele 48

3.2.2. Die Rechenzeiten 49

3.2.3. Die Rechengenauigkeit 62

3.2.4. Die maximale Matrixgrösse 68

3.2.5. Die Konvergenz des Zielfunktionswertes zum Optimalwert 70

4. Schlussbemerkungen 72

5. Flussdiagramm 73 a

Symbolverzeichnis

BV	Basisvariable,
NBV	Nebenbasisvariable,
UP	Urproblem,
TP	Teilproblem,
KOP T	Koordinierungsproblem im Zyklus T
$a_{i,j}$	Koeffizient im UP und in den TP in der Zeile mit Index i und der Spalte mit Index j,
b_i	Nebenbedingungsabsolutbetrag im UP, KOP und in den TP in der Zeile mit Index i,
b_i^*	Wert der Schlupfvariablen der i-ten Nebenbedingung im Optimum des letzten KOP,
c_j	Zielfunktionskoeffizient im UP in der Spalte mit Index j,
$\bar{c}_j^T$	aktueller Wert von c_j nach dem Zyklus T
g_j	Gewichtungsfaktor für die Koeffizienten der j-ten UP-Spalte,
G_k	Anzahl der möglichen, verschiedenen, zulässigen Basislösungen des k-ten TP,
i	Index, der über die Nebenbedingungen des UP läuft,
j	Index, der über die Spalten des UP läuft,
k	Index, der über die Anzahl (Z) aller TP läuft,
m	Anzahl der Nebenbedingungen des UP,

M_{UP}	Menge aller Zeilenindices i des UP,
M_{TP_k}	Menge aller Zeilenindices i des k-ten TP,
n	Anzahl der eigentlichen Variablen des UP (ohne Schlupfvariablen),
N_{UP}	Menge aller Spaltenindices j des UP,
N_{TP_k}	Menge aller Spaltenindices j des k-ten TP,
$P_{t,k}$	Variable eines KOP, die für die optimale Lösung aus dem t-ten Zyklus des k-ten TP definiert wurde,
$\bar{P}^T_{t,k}$	KOP-optimaler Lösungswert für die Variable $P_{t,k}$ im KOP T,
q_T	Anzahl der eigentlichen Variablen des KOP T,
r	Anzahl der eigentlichen Variablen in der optimalen Basis des UP,
t	Index, der über die bereits durchgerechneten Zyklen des Verfahrens bis zum jeweils gerade gegenwärtigen Zyklus T läuft,
T	Index, der angibt, im wievielten Zyklus sich die Rechnung gerade befindet,
u_i	Schlupfvariable für eine Nebenbedingung mit Index i des UP, eines KOP oder TP,
v^{UP}	zu maximierende Zielvariable des UP,
$v^{KOP\ T}$	zu maximierende Zielvariable des KOP T,
$\bar{v}^{KOP\ T}$	KOP-optimaler Wert von $v^{KOP\ T}$,
$v^{TP}_{T,k}$	zu maximierende Zielvariable des k-ten TP im Zyklus T,

$\bar{v}^{TP}_{T,k}$	TP-optimaler Wert von $v^{TP}_{T,k}$,
$\bar{v}^{UP,T}$	Wert der Zielfunktion des UP, der sich für eine zulässige UP-Lösung einstellt, welche der optimalen Primallösung des KOP T entspricht,
x_j	eigentliche Variable des UP mit Index j,
x_j^*	optimaler Wert der UP-Variablen mit Index j,
$\bar{\bar{x}}_j^T$	Wert der UP-Variablen mit Index j aus einer zulässigen UP-Lösung, welche der optimalen Primallösung des KOP T entspricht,
$x_{j,t,k}$	Variable des k-ten TP im Zyklus t, die der j-ten UP-Variablen entspricht,
$\bar{x}_{j,t,k}$	TP-optimaler Lösungswert für die Variable $x_{j,t,k}$,
$\bar{\bar{x}}^T_{j,t,k}$	mit dem optimalen Lösungswert des KOP T : $\bar{P}^T_{t,k}$ gewichteter Wert $\bar{x}_{j,t,k}$,
y_i^{UP}	Dualvariable des UP mit Index i,
$y_i^{KOP\ T}$	Dualvariable des KOP T mit Index i,
$\bar{y}_i^{KOP\ T}$	KOP-optimaler Wert von $y_i^{KOP\ T}$,
Z	Anzahl der TP,
z^{UP}	zu minimierende Zielvariable des dualen UP,
$\bar{z}^{UP}$	optimaler Wert von z^{UP} ,
$z^{KOP\ T}$	zu minimierende Zielvariable des dualen KOP T,
$\bar{z}^{KOP\ T}$	optimaler Wert von $z^{KOP\ T}$,
$\bar{z}^{UP,T}$	$\bar{z}^{UP,T} = \bar{z}^{KOP\ T}$.

1. Vorbemerkungen

1.1. Allgemeines

1.1.1. Mögliche Vorteile eines allgemeinen Dekompositionsverfahrens für lineare Optimierungsprobleme

In dieser Arbeit soll ein allgemeines Dekompositonsverfahren beschrieben werden, mit dem man für jedes beliebige, lineare Optimierungsproblem eine optimale Lösung finden kann, sofern zumindest eine optimale Basislösung überhaupt existiert. Gibt es keine zulässige Lösung oder ist der Bereich zulässiger Lösungen nicht beschränkt, so dass es keinen optimalen Wert der Zielfunktion gibt, so stellt dieses Verfahren auch das fest.

Ein solches Verfahren ist oder könnte in verschiedener Hinsicht von Vorteil sein gegenüber anderen Lösungsmethoden für lineare Optimierungsprobleme :

a) Geringerer Rechenzeitbedarf

Programmiert man dieses Verfahren für eine Datenverarbeitungsanlage, so könnte sich ein geringerer Bedarf an Rechenzeit ergeben als bei anderen Lösungsmethoden. Das ist allerdings nur für grössere Probleme wahrscheinlich, da man bei diesem Verfahren eine sehr viel umfangreichere Programmorganisation programmieren muss, als beispielsweise bei der einfachen Simplex-Methode, so dass bei kleinen Problemen dieser nahezu fixe Organisationsaufwand kaum durch einen schnelleren Rechengang überkompensiert werden kann.

Für jedes Problem kann der Benutzer dieses Verfahrens den Rechenzeitbedarf extern erheblich beeinflussen.

b) Geringerer Kernspeicherbedarf in Rechenanlagen

Dieser Algorithmus hat zwar mehr Daten zu speichern als die meisten anderen Verfahren, jedoch kann das in externen Speichern geschehen, so dass er relativ wenig

Daten im Kernspeicher benötigt, ohne dass die Rechnungen deswegen durch Datenaustausch zwischen externem und internem Speicher besonders häufig unterbrochen werden.

c) Höhere Rechengenauigkeit

Eine grössere Genauigkeit der Rechnung könnte sich ergeben, weil immer wieder auf die Ausgangsdaten zurückgegriffen wird und die maximale Grösse von Matrizen, die als Ganzes zumindest berechnet werden müssen, kleiner gehalten werden kann als bei jedem anderen Lösungsverfahren, welches auf beliebige, lineare Optimierungsprobleme anwendbar ist.

d) Allgemeine Anwendbarkeit des Verfahrens

Im Gegensatz zu fast allen anderen Dekompositionsverfahren, kann der vorliegende Algorithmus auf jedes beliebige, lineare Optimierungsproblem angewendet werden.

1.1.2. Bereits vorhandene Dekompositionsverfahren

1.1.2.1. Spezielle Verfahren

Es existieren bereits eine grosse Anzahl von Dekompositionsverfahren für lineare und nicht-lineare Optimierungsprobleme bei denen eine spezielle Struktur der Koeffizientenmatrix des Problems angenommen wird.[1] Auf diese Verfahren soll hier nicht eingegangen werden.

---

1) Vgl. z.B.: Abadie, J.M., u. A.C. Williams [3], S. 149-158; Balas, E. [6], S. 847-873; Bell, E.J. [11]; Benders, J.F. [13], S. 238-252; Dantzig, G.B., u. P. Wolfe [19], S. 101-111; Heestermann, A.R.G. [24], S. 288-306; Hu, T.C. [27], S. 91-102; Lüttgen, H. [31], S. 370-379; Mills, G. [35], S. 279-291; Müller-Merbach, H. [37], S. 306-322; Rosen, J.B. [42], S. 159-176; Sanders, J.L. [45], S. 266-271; Tan, S.T. [46], S. 168-189 u. 247-268; Whinston, A. [50], S. 188-201; Zangwill, W.I. [51], S. 1068-1087.

1.1.2.2. Generelle Verfahren

An allgemein anwendbaren Dekompositionsverfahren für lineare Optimierungsprobleme ist - von dem vorliegenden Algorithmus abgesehen [2] - bisher nur eines bekannt. Es handelt sich dabei um den Algorithmus von D. Adam und W. Röhrs [3], für den die Autoren keinerlei Annahmen über spezielle Strukturen der Koeffizientenmatrix des Problems machen.

Deren Verfahren (im folgenden Verfahren I genannt) unterscheidet sich in mehreren wichtigen Punkten von dem des Verfassers (im folgenden Verfahren II genannt) :

a) Die nach Verfahren I zu definierenden Unterprobleme müssen bei einer voll besetzten Koeffizientenmatrix im Gegensatz zu Verfahren II alle Nebenbedingungen des Gesamtproblems umfassen [4].

b) Im Verfahren I müssen im Gegensatz zu Verfahren II alle Nebenbedingungs-Absolutbeträge auf die Unterprobleme aufgeteilt werden [4].

c) Nach der dann folgenden Optimierung der Unterprobleme stellt Verfahren I im Gegensatz zu Verfahren II eine Sensitivitätsanalyse an, die für jede optimale Unterproblem-Lösung feststellen soll, inwieweit man die "Kapazitätszuteilung" auf das Unterproblem ändern kann, ohne dass man für die optimale Basis dieses Unterproblems einen Basistausch vornehmen muss, um die Zulässigkeit dieser Basis im Unterproblem zu erhalten. [5]

d) Eine optimale Koordination der Unterproblem-Lösungen kann in Verfahren I auf zwei Arten vorgenommen werden :

(1) In einem Masterproblem, welches explizit die "Kapazitäten" auf die Unterprobleme neu verteilt. Dieses Problem

---

2) s. Heinemann, H., [26], S. 302-317;
3) s. Adam, D., u. W. Röhrs [4], S. 395-417;
Adam, D., [5], S. 196-230;
4) vgl. Adam, D., u. W. Röhrs [4], S. 398;
5) vgl. Adam, D., u. W. Röhrs [4], S. 399f.

hat im allgemeinen Fall m·Z Variable und m·(Z+1) Nebenbedingungen.[6] Dabei ist m die Anzahl der Nebenbedingungen des Gesamtproblems und Z die Anzahl der Teilprobleme.

(2) In ein Master-Problem werden alle Variable, die sich in den optimalen Basen der Unterprobleme befinden, hineingenommen und das Problem dann optimiert. Die Grösse dieses Problems geben Adam und Röhrs mit m·Z Variablen und m Nebenbedingungen an.[7] In Verfahren II kann dieses koordinierende Master-Problem dagegen höchstens m+Z Variablen und m Nebenbedingungen haben.

e) Bei beiden Verfahren ergibt sich aus der optimalen Lösung des Master-Problems eine zulässige Lösung des Gesamtproblems. Jedoch bei Verfahren I (und nur dort) muss dies auch eine Basislösung des Gesamtproblems sein.[8]

f) Eine Aktualisierung der Zielfunktionskoeffizienten des Gesamtproblems wird in beiden Verfahren vorgenommen. Auch werden diese Koeffizienten dann für einen Optimalitätstest und für die Auswahl von neu in die Lösung aufzunehmenden Variablen benutzt. Verfahren I nimmt dabei im Gegensatz zu Verfahren II nie mehr als eine Variable neu auf.[9] Dieses ist jedoch eher ein Unterschied in der rechnerischen Taktik als in der mathematischen Theorie. Diese Punkte deuten schon an, dass zwischen beiden Verfahren ein beachtlicher Unterschied besteht.

---

6) Vgl. Adam, D., und W. Röhrs, [4], S. 402;
7) vgl. Adam, D., und W. Röhrs, [4], S. 406;
8) vgl. Adam, D., und W. Röhrs, [4], S. 407;
9) vgl. Adam, D., und W. Röhrs, [4] S. 396.

1.2. Allgemeine Anwendbarkeit des zu beschreibenden Verfahrens

Das vorliegende Verfahren ist wie die Simplex-Methode gegebenenfalls in zwei Phasen auf jedes lineare Optimierungsproblem anwendbar. Insbesondere werden keine Annahmen über bestimmte Besetzungsmuster der Koeffizientenmatrix eines Problems gemacht.

Lediglich zum Zwecke der einfacheren Darstellung wird hier angenommen :

a) Es sind nur stets Maximierungsprobleme zu lösen.
b) Für das Problem ist bereits eine primal zulässige Basislösung als Ausgangslösung bekannt.

Dadurch wird die allgemeine Anwendbarkeit des Verfahrens nicht eingeschränkt. Denn erstens kann man jede zu minimierende Zielfunktion in eine zu maximierende überführen, indem man alle ihre Koeffizienten mit dem Faktor (-1) multipliziert. Zum anderen liegt eine zulässige Ausgangsbasislösung manchmal schon vor, wenn man allen Schlupfvariablen den Absolutbetrag der zugehörigen Ungleichung als Wert zuweist. In allen anderen Fällen kann man das Problem durch Einführung künstlicher Variablen in ein nicht äquivalentes Problem mit in bezug auf das neue Problem zulässiger Ausgangsbasislösung verwandeln. Durch Lösung des neuen Problems mit Hilfe einer geeigneten Zielfunktion kann man stets eine zulässige Basislösung des ursprünglichen Problems gewinnen, sofern eine solche überhaupt existiert. Auch zur Lösung dieses nicht äquivalenten Problems kann das Dekompositionsverfahren benutzt werden.

Daher kann man das Verfahren in dem Sinne als primal bezeichnen, dass in Fällen ohne zulässige Ausgangsbasislösung zunächst eine zulässige, primale und dann eine optimale, primale Lösung gesucht wird.

2. Ein generelles Dekompositionsverfahren für lineare Optimierungsprobleme

2.1. Kurze Skizzierung des Verfahrens

Zur Erleichterung der weiteren Lektüre sei das Verfahren vorweg in seinen Hauptpunkten kurz skizziert :

(1) Aus dem zu lösenden Gesamtproblem werden Teilprobleme gebildet.

(2) Die Teilprobleme werden mit einer geeigneten Zielfunktion versehen und dann jedes für sich optimiert.

(3) Die Lösungen der Teilprobleme dienen der Gewinnung von Variablen für ein Hilfsproblem, welches anstelle des Gesamtproblems optimiert wird und die bereits vorhandenen Teilproblem-Lösungen optimal kombiniert.

(4) Mit Hilfe der optimalen Dualwerte des Hilfsproblems errechnet man aktualisierte Zielfunktionswerte für das zu lösende Gesamtproblem.

(5) Mit diesen aktualisierten Zielfunktionswerten testet man, ob das Optimum des Gesamtproblems bereits erreicht ist.

(6) Ist das Optimum erreicht, so ist das Verfahren beendet. Andernfalls setzt man das Verfahren unter Verwendung der aktualisierten Zielfunktionswerte aus Punkt (3) mit Punkt (2) fort, womit ein neuer Zyklus des Verfahrens beginnt.

Der Algorithmus ermöglicht verschiedene Strategien bei seiner Anwendung. Je nach Art des Problems und je nach angewandter Rechen-Strategie ähnelt das Verfahren dann mehr der revidierten Simplex-Methode oder dem Dekompositionsverfahren von Dantzig und Wolfe [10].

---

10) Vgl. Dantzig, G.B., und P. Wolfe [19], S. 101-111.

2.2. Genaue Beschreibung des Verfahrens

Durch Dekomposition zu lösen sei folgendes lineare Optimierungsproblem, welches im folgenden als Urproblem oder als UP bezeichnet wird :

Maximiere

$$\sum_{j=1}^{n} c_j \cdot x_j = v^{UP}$$

unter Beachtung der Nebenbedingungen

$$\sum_{j=1}^{n} a_{i,j} \cdot x_j \leq b_i \qquad \text{für } i = 1, 2, \ldots, m$$

$$x_j \geq 0 \qquad \text{für } j = 1, 2, \ldots, n \quad , \tag{1}$$

wobei alle b_i nicht-negativ sein sollen.

Dabei bezeichnet :

x_j : eine Variable des UP mit dem Index j,

j : einen über alle Variablen des UP laufenden Index,

n : die Anzahl der eigentlichen UP-Variablen,

c_j : den Zielfunktionskoeffizienten der j-ten UP-Variablen,

v^{UP} : die primale Zielvariable,

$a_{i,j}$: den Koeffizienten der j-ten Variablen in der i-ten Nebenbedingung des UP,

i : einen über alle Nebenbedingungen des UP laufenden Index,

m : die Anzahl der Nebenbedingungen des UP ohne Nicht-Negativitätsbedingungen und

b_i : den Absolutbetrag der i-ten Nebenbedingung des UP.

Zur Veranschaulichung soll das Dekompositionsverfahren an folgendem, in Tableauform dargestellten Zahlenbeispiel demonstriert werden, wobei die u die Schlupfvariablen der

Restriktionen sind und die anfängliche Basislösung $u_i = b_i$ für $i = 1, 2, 3$ ist.

Tabelle 1 : Das UP

BV \ NBV	x_1	x_2	x_3	x_4	x_5	x_6	x_7	Absolutbeträge
$-v^{UP}$	5	6	5	4	5	6	8	0
u_1	4	3	4	0	6	3	5	200
u_2	3	6	3	-5	1	3	6	180
u_3	2	6	0	4	4	2	4	200

BV : Basisvariable
NBV : Nebenbasisvariable

2.2.1. Bildung von Teilproblemen

Wie bereits angedeutet, wird anstelle des UP in jedem Zyklus ein koordinierendes Hilfsproblem berechnet, welches im folgenden als Koordinierungsproblem oder kurz als KOP bezeichnet wird.

Die Bildung von Teilproblemen (im folgenden TP genannt) und deren spätere Optimierung dient fast ausschliesslich dem Zwecke, für das KOP Variablen zu beschaffen. Appriori nicht ganz auszuschliessen ist allerdings die Möglichkeit, dass es hierbei auch darum geht, "besonders geeignete" Variablen für das KOP zu gewinnen. Sieht man von letzterer Möglichkeit jedoch ab, so könnte man die Bildung und Optimierung der TP in diesem Algorithmus theoretisch ebenso durch andere Verfahren zur Gewinnung von KOP-Variablen ersetzen.

In der vorliegenden Form des Algorithmus geht man wie folgt vor :

a) man bildet TP;
b) man versieht die TP mit einer Zielfunktion;
c) man optimiert die TP jedes für sich;
d) aus den so erhaltenen TP-Lösungen leitet man Koeffizienten für die KOP-Variablen ab.

Schritt I des Algorithmus :

Man bilde eine beliebige Anzahl Z, jedoch mindestens ein TP, wobei das k-te TP (für k = 1,...,Z) gleich mit einer geeigneten Zielfunktion versehen wie folgt zu schreiben ist :

Maximiere

$$\sum_{j \in N_{TP_k}} \left(c_j - \sum_{i=1}^{m} a_{i,j} \cdot \bar{y}_i^{KOP\ T-1}\right) \cdot x_{j,T,k} = v_{T,k}^{TP}$$

unter Beachtung der Nebenbedingungen (2)

$$\sum_{j \in N_{TP_k}} a_{i,j} \cdot x_{j,T,k} \leq b_i \qquad (i \in M_{TP_k})$$

$$x_{j,T,k} \geq 0 \qquad (j \in N_{TP_k}),$$

wobei gelten muss :

$$j \in N_{TP_1} \cup N_{TP_2} \cup \ldots\ldots \cup N_{TP_Z} .$$

Dazu ist zu erklären, dass das Verfahren in Zyklen eingeteilt ist, die jeweils damit beginnen, dass man in die einmal zu Beginn des Verfahrens eingeteilten TP eine aktuelle Zielfunktion einsetzt, wobei der jeweils gerade laufende Zyklus stets mit dem Index T und der vorhergehende mit dem Index T-1 gekennzeichnet wird.

Die $\bar{y}_i^{KOP\ T-1}$ (für i=1,...,m) sind optimale Dualwerte, die sich im jeweiligen Zyklus T-1 für die i-te Nebenbedingung des KOP T-1 eingestellt haben. Da das Verfahren mit dem

Zyklus T=1 beginnt, existiert kein Zyklus (T-1) = 0. Daher gilt für den ersten Zyklus T die Definition :

$$\bar{y}_i^{\text{KOP T-1}} = \bar{y}_i^{\text{KOP 0}} = 0 \qquad \text{für } i = 1,\ldots,m.$$

Die $\bar{y}_i^{\text{KOP T-1}}$ seien dabei so definiert, dass sie im Optimum des KOP T-1 nicht-negative Werte aufweisen.

Ferner gelten folgende Definitionen :

a) Es sei N_{UP} (M_{UP}) die Menge aller Spalten-Indices j (Zeilen-Indices i) des UP.

b) Davon sei N_{TP_k} (M_{TP_k}), die Menge aller Spalten-Indices j (Zeilen-Indices i) des k-ten TP (k=1,...,Z), eine Teilmenge :

$$N_{TP_k} \subset N_{UP} \qquad (k=1,\ldots,Z) \qquad (3)$$

und

$$M_{TP_k} \subset M_{UP} \qquad (k=1,\ldots,Z). \qquad (4)$$

Bei der Bildung der TP ist zu beachten, dass jede UP-Variable mit Index j zumindest in einem TP vorkommt, dass also jeder Index j ein Element der Vereinigung aller Mengen N_{TP_k} ist. Würde das für ein bestimmtes j nicht gelten, so könnte die betreffende UP-Variable x_j bei diesem Verfahren niemals in eine UP-Lösung gelangen.

Andererseits ist es nicht notwendig, dass jeder Nebenbedingung des UP in zumindest einem TP eine Nebenbedingung entspricht. Es braucht also nicht jeder Index i ein Element der Vereinigung aller Mengen M_{TP_k} zu sein, da die Berücksichtigung aller Nebenbedingungen des UP später bei der Bildung des KOP sichergestellt wird, wie unten noch gezeigt wird, Im Zahlenbeispiel ist z.B. die dritte Nebenbedingung in keinem TP enthalten.

Ebensowenig ist es notwendig, dass der Durchschnitt aller Mengen N_{TP_k} oder M_{TP_k} leer ist. Im ersten Falle könnten die

betreffenden UP-Variablen über alle TP in das KOP gelangen. Es ist fraglich, ob eine solche TP-Einteilung rechentechnisch sinnvoll ist, aber das Auffinden des UP-Optimums würde dadurch nicht beeinträchtigt. Im zweiten Falle wären die betreffenden Nebenbedingungen in allen TP vorhanden, wie es in dem gewählten Zahlenbeispiel für i = 1, 2 geschieht. Es kann nur durch praktische Rechnungen geklärt werden, ob dieses Vorgehen rechentechnisch vorteilhaft ist.

Es sei hier schon angemerkt, dass die im ersten Zyklus definierten Mengen N_{TP_k} und M_{TP_k} (k=1,...,Z) in den folgenden Zyklen nicht umdefiniert werden, sondern unverändert über alle Zyklen gelten. Eine Umdefinition dieser Mengen in späteren Zyklen wäre zwar zulässig, wenn weiterhin gewährleistet ist, dass gilt :

$j \in N_{TP_1} \cup N_{TP_2} \cup \ldots \cup N_{TP_Z}$. Jedoch gibt es nach Ansicht des Verfassers keinen Grund, von dieser Möglichkeit Gebrauch zu machen.

Bezugnehmend auf das Zahlenbeispiel sollen zwei TP gebildet werden (Z = 2) :

TP 1 : $M_{TP_1} = \{1,2\}$ $N_{TP_1} = \{1,2,3,4\}$

Tabelle 2

TP 1 zu Beginn des ersten Zyklus

BV \ NBV	$x_{1,1,1}$	$x_{2,1,1}$	$x_{3,1,1}$	$x_{4,1,1}$	Absolutbeträge
$-v^{TP}_{1,1}$	5	6	5	4	0
u_1	4	3	4	0	200
u_2	3	6	3	-5	180

TP 2 : $M_{TP_2} = \{1,2\}$ $N_{TP_2} = \{5,6,7\}$

Tabelle 3

TP 2 zu Beginn des ersten Zyklus

BV \ NBV	$x_{5,1,2}$	$x_{6,1,2}$	$x_{7,1,2}$	Absolut-beträge
$-v^{TP}_{1,2}$	5	6	8	0
u_1	6	3	5	200
u_2	1	3	6	180

Die Aufteilung der TP ist hier so gewählt, dass der Index $i = 3$ nicht Element der Vereinigung der Mengen M_{TP_1} und M_{TP_2} ist.

2.2.2. Lösen der Teilprobleme

Schritt II des Algorithmus :

Man suche für jedes TP eine TP-optimale Lösung.

Bei der Ausführung dieses Schrittes sind zwei Schwierigkeiten denkbar, die beide dazu führen können, dass die TP nicht ihre Aufgabe erfüllen können, jeder UP-Variablen bei Bedarf irgendwie Eingang in das KOP zu verschaffen. Träfe man für diese Fälle nicht in geeigneter Weise Vorsorge, so würde das UP-Optimum u.U. nicht erreicht.

Die erste Schwierigkeit kann bei einem degenerierten TP auftreten, wenn im k-ten TP z.B. zumindest für ein i mit $i \in M_{TP_k}$ gilt : $b_i = 0$. Dann existiert für dieses TP möglicherweise

- entweder überhaupt keine TP-zulässige Lösung (von $u_i = b_i$ für $i \in M_{TP_k}$ abgesehen), wenn für das betreffende i nämlich gilt : $a_{i,j} > 0$ für alle j mit $j \in N_{TP_k}$. Dann könnte überhaupt keine Variable aus diesem TP in ein KOP gelangen.

- Oder es kann u.U. eine einzelne Variable mit $s \in N_{TP_k}$ nicht in eine TP-zulässige Lösung des k-ten TP kommen, wenn für das betreffende i gilt : $a_{i,s} > 0$ und $a_{i,j} \geq 0$ für $j \neq s$ und alle $j \in N_{TP_k}$.

Das Auftreten dieses Falles kann jedoch sehr einfach verhindert werden, indem man in kein TP Nebenbedingungen mit $b_i = 0$ aufnimmt, denn die Tatsache, dass Nebenbedingungen in keinem TP vertreten sind, ist völlig unschädlich, da später das KOP eine Lösung herstellt, die einer zulässigen UP-Lösung entspricht.

Möchte man den Benutzer dieses Algorithmus davon befreien, bei der Einteilung der TP auf diese möglichen Schwierigkeiten achten zu müssen, dann kann man bei der Programmierung dieses Verfahrens programmintern geeignete Vorsorgemassnahmen treffen, die gewährleisten, dass auch bei Auftreten dieser Komplikation das UP-Optimum gefunden wird, sofern es überhaupt existiert. Man braucht nur zu gewährleisten, dass in jedem Falle jede UP-Variable in das KOP gelangen kann. Das kann beispielsweise geschehen, indem man die Anzahl der TP um eins erhöht ($Z := Z + 1$) und in jedem Zyklus T das folgende "künstliche" TP mit Index Z mitberechnet :

Maximiere

$$\sum_{j=1}^{n} (c_j - \sum_{i=1}^{m} a_{i,j} \cdot \bar{y}_i^{KOP\ T-1}) \cdot x_{j,T,Z} = v_{T,Z}^{TP} \qquad (5)$$

unter Beachtung der Nebenbedingungen

$$\sum_{j=1}^{n} x_{j,T,Z} \leq B_Z \qquad (B_Z > 0)$$

$$x_{j,T,Z} \geq 0 \qquad (j = 1,\ldots,n).$$

B_Z ist dabei eine beliebig grosse, positive Konstante, die

im Testprogramm des Verfassers für dieses Verfahren einfach gleich 2 gesetzt wurde. In diesem "künstlichen" TP sind alle UP-Variablen enthalten und jede kann bei Bedarf stets in ein KOP gelangen.

Man könnte diesen Algorithmus auch rechnen, wenn man ausser diesem "künstlichen" TP keine anderen TP definiert, denn die Hauptaufgabe der TP, die benötigten Variablen für das KOP auszuwählen, kann dieses TP schon erfüllen. Dieses Vorgehen hätte nur den erheblichen Nachteil, dass man für jeden Basistausch, den die normale Simplex-Methode machen würde, mit diesem Algorithmus einen ganzen Zyklus machen müsste.

Die zweite der erwähnten möglichen Komplikationen bei der Lösung der TP ist, dass der Zielfunktionswert eines TP gegen unendlich gehen kann und man daher keine TP-optimale TP-Lösung findet.

Für diesen Fall kann man verschiedene Gegenmassnahmen treffen. Einmal könnte man ein "künstliches" TP wie unter (5) auch für diesen Fall einführen. Dann kann - wie oben bereits erwähnt - auf alle weiteren TP und damit auch auf deren optimale Lösungen verzichtet werden, da die Erzeugung der benötigten KOP-Variablen durch diesen Notbehelf bereits gewährleistet ist.

Man kann aber auch so vorgehen, dass man vorsichtshalber für jedes TP mit Index k eine künstliche, zusätzliche Restriktion folgender Art einführt, die den Bereich TP-zulässiger Lösungen des TP in jedem Falle begrenzt :

$$\sum_{j \in N_{TP_k}} x_{j,T,k} \leq B_k \qquad (B_k >> 0) \qquad (6)$$

B_k könnte dabei so gross gewählt werden, dass diese Restriktion nur wirksam wird, wenn der Zielfunktionswert des TP in Abwesenheit dieser Restriktion gegen unendlich gehen würde.

Wenn man der Ansicht ist, dass der Fall gegen unendlich gehender Werte der TP-Zielfunktion nicht so häufig auftritt, dass sich seine vorbeugende Behandlung lohnt, zumal man auch bei der Einteilung der TP darauf Einfluss nehmen kann, dann kann man auch wie folgt vorgehen :

Droht bei der Berechnung eines TP beim nächsten Basistausch der Wert seiner Zielfunktion gegen unendlich zu gehen, so führe man noch vor diesem Basistausch in das aktuelle TP-Tableau eine neue Zeile und eine neue Schlupfvariable als Basisvariable derart ein, dass in dieser Zeile jede andere Basisvariable einen Koeffizienten von null, jede Nebenbasisvariable einen positiven Koeffizienten hat und der Nebenbedingungsabsolutbetrag ebenfalls positiv ist. Damit ist dann das Anwachsen des Wertes der TP-Zielfunktion begrenzt.

Die TP-optimale Lösung (gegebenenfalls unter Berücksichtigung einer zusätzlichen, künstlichen Restriktion) eines k-ten TP aus dem t-ten Zyklus besteht aus einem Vektor, dessen Elemente wie folgt bezeichnet werden :

$$\bar{x}_{j,t,k} \qquad (j \in N_{TP_k}) \qquad (7)$$

Bezüglich des Beispiels ergibt sich damit folgendes :

Für TP 1 ergibt sich im Verlaufe der Rechnung das in Tabelle 4 dargestellte Tableau, in dem die dritte Zeile wie oben beschrieben hinzugefügt wurde, weil der Wert der Zielfunktion beim nächsten Basistausch, durch den die Schlupfvariable u_2 in die Basis kommt, gegen unendlich zu gehen drohte.

Tabelle 4 : TP 1 im Verlaufe des ersten Zyklus

BV \ NBV	$x_{1,1,1}$	u_2	$x_{3,1,1}$	u_1	Absolut-beträge
$-v^{TP}_{1,1}$	-7	+0,8	-7	-3,6	-576
$x_{4,1,1}$	+1	-0,2	+1	+0,4	+ 44
$x_{2,1,1}$	+4/3	+0,0	+4/3	+1/3	+ 66 2/3
u_3	+1	+1	+1	+1	+500

Damit ergeben sich folgende TP-Lösungen im ersten Zyklus :

TP 1: $\bar{x}_{2,1,1}$ = 66 2/3;

$\bar{x}_{4,1,1}$ = 144;

$\bar{x}_{1,1,1} = \bar{x}_{3,1,1}$ = 0 und

$\bar{v}^{TP}_{1,1}$ = 976 als Zielfunktionswert.

TP 2: $\bar{x}_{5,1,2}$ = 4;

$\bar{x}_{6,1,2}$ = 58 2/3;

$\bar{x}_{7,1,2}$ = 0 und

$\bar{v}^{TP}_{1,2}$ = 372 als Zielfunktionswert.

2.2.3 Aufstellung eines Koordinierungsproblems (KOP)

Die bisher errechneten TP-Lösungen sind nun in einer Art und Weise weiterzuverwenden, die einer Annäherung an das

Optimum des UP entspricht. Das geschieht im KOP, welches durch optimale Gewichtung der errechneten TP-Lösungen zur bestmöglichen, UP-zulässigen und daraus ableitbaren UP-Lösung führt.

Jeder im laufenden Zyklus T gewonnenen TP-Lösung, die keine Nullpunktlösung ist, entspricht dabei eine KOP-Variable. Die Koeffizientenspalte für eine jede KOP-Variable wird dabei aus der entsprechenden TP-Lösung und den Koeffizienten des UP wie folgt errechnet :

a) Der Zielfunktionskoeffizient der KOP-Variablen, die der Lösung des k-ten TP aus dem laufenden Zyklus (t=T) entspricht, ergibt sich als :

$$\sum_{j \in N_{TP_k}} c_j \cdot \bar{x}_{j,t,k} \qquad (8)$$

Dieser Koeffizient ist für alle im laufenden Zyklus T errechneten TP-Lösungen zu berechnen (k=1,..,Z). Er gibt an, welchen "Gewinn" die betreffende TP-Lösung brächte, wenn man sie realisieren könnte und würde.

b) Entsprechend ist für die Lösung eines jeden TP mit Index k (k=1,..,Z) aus dem laufenden Zyklus (t=T) ein Spaltenvektor mit den folgenden m Elementen zu berechnen:

$$\sum_{j \in N_{TP_k}} a_{i,j} \cdot \bar{x}_{j,t,k} \quad \text{für} \quad (i=1,..,m). \qquad (9)$$

Der i-te Koeffizient dieses Vektors gibt an, in welchem Umfang die betreffende TP-Lösung die i-te UP-Restriktion beanspruchen würde, wenn man diese TP-Lösung voll realisieren würde.

Jede dieser Koeffizientenspalten (t=T; k=1,..,Z) ist eine Linearkombination aus den Koeffizientenspalten des UP mit den nicht-negativen Gewichten

$$\bar{x}_{j,t,k} \qquad (j \in N_{TP_k}).$$

Die Variablen des KOP werden mit $P_{t,k}$ bezeichnet und können als Gewichtungsfaktoren für die TP-Lösungen verstanden werden.

Bezogen auf das Beispiel ergeben sich im ersten Zyklus (T=1) folgende KOP-Koeffizienten :

für TP 1 (KOP-Variable $P_{1,1}$)

für (8): $6 \cdot 66\ 2/3 + 4 \cdot 144 = 976$
für (9): $3 \cdot 66\ 2/3 + 0 \cdot 144 = 200$
$6 \cdot 66\ 2/3 + (-5) \cdot 144 = -320$
$6 \cdot 66\ 2/3 + 4 \cdot 144 = 976$

für TP 2 (KOP-Variable $P_{1,2}$)

für (8): $5 \cdot 4 + 6 \cdot 58\ 2/3 = 372$
für (9): $6 \cdot 4 + 3 \cdot 58\ 2/3 = 200$
$1 \cdot 4 + 3 \cdot 58\ 2/3 = 180$
$4 \cdot 4 + 2 \cdot 58\ 2/3 = 133\ 1/3$

In diesem Beispiel ist die Lösung des TP 1 im ersten Zyklus keine zulässige Lösung des UP, da dessen dritte Restriktion verletzt würde (976 würden beansprucht, 200 stehen nur zur Verfügung), wenn man diese Lösung voll realisieren würde.

Über die im jeweils laufenden Zyklus T neu gewonnenen KOP-Variablen hinaus gehen noch weitere Variable in das KOP ein; nämlich jene, die im vorhergehenden Zyklus T-1 in der optimalen Basis des betreffenden KOP T-1 waren.

Lediglich im Zyklus T=1 kann es solche Variablen noch nicht geben, weil ein Zyklus T=0 nicht existiert.

Damit kann man das KOP T für den Zyklus T folgendermassen schreiben, wobei hier der Einfachheit halber auch die Variablen für TP-Lösungen, die Nullpunktlösungen wurden, und solche Variable, die nicht in der optimalen Basis des je-

weiligen KOP T-1 waren, mit aufgeführt werden :

Maximiere

$$v^{KOP\ T} = \sum_{t=1}^{T} \sum_{k=1}^{Z} \left(\sum_{j \in N_{TP_k}} c_j \cdot \bar{x}_{j,t.k} \right) \cdot P_{t,k} \qquad (10)$$

unter Beachtung der Nebenbedingungen

$$\sum_{t=1}^{T} \sum_{k=1}^{Z} \left(\sum_{j \in N_{TP_k}} a_{i,j} \cdot \bar{x}_{j,t,k} \right) \cdot P_{t,k} \leq b_i$$

$$(i = 1,\ldots,m)$$

$$P_{t,k} \geq 0 \quad (t = 1,\ldots,T;\ k = 1,\ldots,Z).$$

Stets hat ein KOP - so wie es bisher beschrieben wurde - m Nebenbedingungen. Das sind genausoviele wie im UP. Im ersten Zyklus kann das KOP höchstens Z Variablen haben, denn mehr TP sind nicht vorhanden und vorhergehende Zyklen, aus denen man noch Variablen hätte übernehmen können, existieren nicht.

Wie bereits angedeutet, werden in jedes KOP T jene Variablen aus dem KOP T-1 übernommen, die in dessen optimaler Basis waren, um zu gewährleisten, dass das KOP T mindestens eine genausogute Lösung hat wie das KOP T-1. Alle Nebenbasisvariablen des KOP-T-Tableaus werden gestrichen, um ein zu grosses Anwachsen der Anzahl der KOP-Variablen zu verhindern. Aus diesem Vorgehen ergibt sich, dass kein KOP mehr als m+Z Variablen haben kann, denn maximal m können in der optimalen Basis des vorhergehenden KOP gewesen sein und maximal Z können im laufenden Zyklus von allen TP dazugekommen sein.

Somit lautet :

Schritt III des Algorithmus :

Man bilde in Zyklus T ein KOP T wie in (10) aus allen aus der optimalen Basis des KOP T-1 übernommenen Variablen und aus den gemäss (8) und (9) im Zyklus T neu gebildeten Variablen.

In Tabelle 5 ist für das Beispiel das Ausgangstableau für das KOP 1 dargestellt :

Tabelle 5 Ausgangstableau des KOP 1

NBV / BV	$P_{1,1}$	$P_{1,2}$	Absolutbeträge
$-v^{KOP\ 1}$	+976	+372	0
u_1	+200	+200	+200
u_2	-320	+180	+180
u_3	+976	+133 1/3	+200

2.2.4 Lösen des Koordinierungsproblems (KOP)

Schritt IV des Algorithmus :

Für das KOP T suche man eine optimale Lösung.

Zur Ausführung dieses Schrittes kann man sich der normalen oder der revidierten Simplex-Methode oder dieses Dekompositionsalgorithmus selbst bedienen. Die revidierte Simplex-Methode scheint aber a priori weniger vorteilhaft zu sein. Denn es kann in einem KOP die Anzahl der Variablen maximal um Z grösser sein als die Anzahl der Restriktionen. Für solche Fälle rentiert sich die Anwendung der revidierten Simplex-Methode i.a. aber nicht. A priori lässt sich auch kaum angeben, ob es vorteilhafter ist, das KOP jeweils von Grund auf neu zu lösen oder jeweils von der optimalen Basis des vorhergehenden KOP auszugehen, denn in jedem Zyklus

kann die Anzahl der neu in das KOP aufgenommenen Variablen bis maximal auf Z steigen. Unter welchen Umständen dennoch die revidierte Simplex-Methode oder dieser Dekompositionsalgorithmus mit Vorteil gegenüber der normalen Simplex-Methode zur Lösung des KOP eingesetzt werden können, wird an späterer Stelle noch besprochen.

Die primale Optimal-Lösung des KOP T ist ein Vektor mit den T mal Z Elementen

$$\bar{P}^{T}_{t,k} \qquad (t=1,..,T;\ k=1,..,Z). \qquad (11)$$

Bei tatsächlichen Rechnungen hat dieser Vektor für T>1 i.a. allerdings weniger als T mal Z Elemente, weil immer wieder KOP-Variable gestrichen werden und somit auch das entsprechende Element aus diesem Vektor gestrichen werden kann. Ausserdem gilt besonders für spätere Zyklen einer Rechnung, dass nicht mehr alle TP in einem Zyklus eine vom Nullpunkt verschiedene optimale Lösung haben und dafür somit auch keine KOP-Variable definiert zu werden braucht, da diese als Koeffizientenvektor nur einen Nullvektor hätte.

Diese KOP-Variablen oder Gewichte für die TP-Lösungen können in einem KOP-Optimum durchaus Werte annehmen, die grösser als 1 sind, wie das Zahlenbeispiel noch zeigen wird.

Die duale Optimal-Lösung des KOP T sei ein Vektor mit den folgenden m Elementen :

$$\bar{y}_i^{KOP\ T} \qquad \text{für} \qquad i=1,..,m. \qquad (12)$$

Für das Zahlenbeispiel ergibt sich im Optimum des KOP 1 folgendes :

a) Wert der Zielfunktion :

$$\bar{v}^{KOP\ 1} \simeq 419{,}78481;$$

b) Primallösung :

$\bar{P}^{1}_{1,1} \simeq 0{,}07911$ und

$\bar{P}^{1}_{1,2} \simeq 0{,}92089;$

c) Duallösung :

$\bar{y}^{KOP\ 1}_{1} \simeq + 1{,}38215$

$\bar{y}^{KOP\ 1}_{2} = 0$ und

$\bar{y}^{KOP\ 1}_{3} \simeq + 0{,}71677.$

Zu fragen bleibt hier, was man mit der Berechnung einer KOP-Lösung erreicht hat. Das sind zwei Dinge. Erstens kann man aus der primalen, optimalen KOP-Lösung eine primale, UP-zulässige UP-Lösung ableiten. Zweitens kann man mit Hilfe der dualen, optimalen KOP-Lösung Kriterien ableiten, mit denen man feststellen kann, ob die abgeleitete UP-Lösung schon UP-optimal ist, oder, wie man eine KOP-Lösung im nächsten Zyklus finden kann, aus der sich eine verbesserte UP-Lösung ableiten lässt.

Zunächst sei aufgestellt die

Behauptung I

Aus einer optimalen Primallösung des KOP T :

$$\begin{array}{l} \bar{P}^{T}_{1,1}\ ,\ \bar{P}^{T}_{1,2}\ ,\ \ldots\ ,\ \bar{P}^{T}_{1,Z} \\ \bar{P}^{T}_{2,1}\ ,\ \bar{P}^{T}_{2,2}\ ,\ \ldots\ ,\ \bar{P}^{T}_{2,Z} \\ \quad\vdots \qquad\qquad\quad \vdots \\ \bar{P}^{T}_{T,1}\ ,\ \bar{P}^{T}_{T,2}\ ,\ \ldots\ ,\ \bar{P}^{T}_{T,Z} \end{array}$$

mit dem zugehörigen Wert der KOP-Zielfunktion $\bar{v}^{KOP\ T}$ lässt sich eine zugehörige, zulässige Primal-Lösung des UP ableiten mit einem Zielfunktionswert des UP von $\bar{v}^{UP,T} = \bar{v}^{KOP\ T}$.

Beweis :

Die optimale Lösung des KOP T erfüllt alle m KOP-Nebenbedingungen :

$$\sum_{t=1}^{T} \sum_{k=1}^{Z} \Big(\sum_{j \in N_{TP_k}} a_{i,j} \cdot \bar{x}_{j,t,k} \Big) \cdot \bar{P}^{T}_{t,k} \leq b_i \qquad (13)$$

$$(i=1,\ldots,m).$$

Definiert man jetzt :

$$\bar{\bar{x}}^{T}_{j,t,k} := \bar{x}_{j,t,k} \cdot \bar{P}^{T}_{t,k} \qquad (14)$$

$$(t=1,\ldots,T; \quad k=1,\ldots,Z; \quad j \in N_{TP_k}),$$

bringt man die TP-Lösungen $\bar{x}_{j,t,k}$ also durch Multiplikation mit den ermittelten Gewichten $\bar{P}^{T}_{t,k}$ auf ihr aktuelles Niveau $\bar{\bar{x}}^{T}_{j,t,k}$, so kann man (13) auch wie folgt schreiben :

$$\sum_{t=1}^{T} \sum_{k=1}^{Z} \sum_{j \in N_{TP_k}} a_{i,j} \cdot \bar{\bar{x}}^{T}_{j,t,k} \leq b_i \qquad (15)$$

$$(i=1,\ldots,m).$$

Definiert man weiterhin :

$$\bar{\bar{x}}^{T}_{j} := \sum_{t=1}^{T} \sum_{k=1}^{Z} \bar{\bar{x}}^{T}_{j,t,k} \qquad (16)$$

$$(j=1,\ldots,n),$$

was für jeden Index j einer Addition der aktualisierten TP-Lösungswerte mit Index j über alle TP und bisherigen Zyklen entspricht - wobei jeweils für bestimmte k für alle $j \notin N_{TP_k}$ $\bar{\bar{x}}^{T}_{j,t,k} = 0$ unterstellt wird -, so kann man (15) wiederum umschreiben in :

$$\sum_{j=1}^{n} a_{i,j} \cdot \bar{\bar{x}}_j^T \leq b_i \qquad (17)$$

$$(i=1,\ldots,m).$$

Das bedeutet, dass die aus einer optimalen Lösung des KOP T ableitbaren $\bar{\bar{x}}_j^T$ eine zulässige Lösung des UP bilden. Dabei müssen die $\bar{\bar{x}}_j^T$ nicht unbedingt eine Basislösung des UP bilden, wie das Zahlenbeispiel noch zeigen wird.

Der dieser UP-Lösung zugehörige Wert der UP-Zielfunktion ist :

$$\bar{v}^{UP,T} = \sum_{j=1}^{n} c_j \cdot \bar{\bar{x}}_j^T \quad . \qquad (18)$$

Der der Lösung des KOP mit den Vektor-Elementen $\bar{P}_{t,k}^T$ $(t=1,\ldots,T;\ k=1,\ldots,Z)$ zugehörige Zielfunktionswert des KOP T ist dagegen :

$$\bar{v}^{KOP\ T} = \sum_{t=1}^{T} \sum_{k=1}^{Z} \Big(\sum_{j \in N_{TP_k}} c_j \cdot \bar{x}_{j,t,k} \Big) \cdot \bar{P}_{t,k}^T \quad . \qquad (19)$$

Formt man (19) gemäss den Definitionen (14) und (16) analog (15) und (17) um, so erhält man :

$$\bar{v}^{KOP\ T} = \sum_{j=1}^{n} c_j \cdot \bar{\bar{x}}_j^T \quad . \qquad (20)$$

Aus (18) und (20) folgt :

$$\bar{v}^{UP,T} = \bar{v}^{KOP\ T} \quad . \qquad (21)$$

Damit ist Behauptung 1 bewiesen.

Für den Sonderfall, dass der Wert der Zielfunktion eines beliebigen KOP T gegen unendlich geht, folgt aus (21), dass dann der Zielfunktionswert der zugehörigen UP-Lösung $\bar{v}^{UP,T}$ gegen unendlich geht. Dann ist das Verfahren beendet, weil es zwar zulässige, primale Lösungen, aber keine Optimal-Lösung des UP gibt.

Umgekehrt gilt auch :

Behauptung II :

> Existiert für den Zielfunktionswert des UP keine obere Schranke, so ergibt sich nach einer endlichen Anzahl von Verfahrenszyklen ein KOP T, für dessen Zielfunktionswert ebenfalls keine obere Schranke existiert, vorausgesetzt, dass etwaig auftretende Degeneration so behandelt wird, dass sich optimale KOP-Basis-Lösungen als solche nicht wiederholen können.

Der Beweis dieser Behauptung erfolgt erst in Abschnitt 2.3.

Übertragen auf das Zahlenbeispiel ergibt sich zunächst für Relation (14) :

für TP 1 :

$\bar{\bar{x}}^{\wedge}_{1,1,1} = 0 \cdot 0{,}07911 = 0$,

$\bar{\bar{x}}^{\wedge}_{2,1,1} \approx 66\ 2/3 \cdot 0{,}07911 \approx 5{,}27400$,

$\bar{\bar{x}}^{\wedge}_{3,1,1} = 0 \cdot 0{,}07911 = 0$ und

$\bar{\bar{x}}^{\wedge}_{4,1,1} \approx 144 \cdot 0{,}07911 \approx 11{,}39144$.

für TP 2 :

$\bar{\bar{x}}^{\wedge}_{5,1,2} \approx 4 \cdot 0{,}92089 = 3{,}68356$,

$\bar{\bar{x}}^{\wedge}_{6,1,2} \approx 58\ 2/3 \cdot 0{,}92089 \approx 54{,}02514$ und

$\bar{\bar{x}}^{\wedge}_{7,1,2} = 0 \cdot 0{,}92089 = 0$.

Für Relation (16) ergibt sich in diesem einfachen Fall

ohne weitere Summation (im ersten Zyklus ist T=1) :

$\bar{\bar{x}}_1^1 = 0$,

$\bar{\bar{x}}_2^1 \simeq 5{,}27400$,

$\bar{\bar{x}}_3^1 = 0$,

$\bar{\bar{x}}_4^1 \simeq 11{,}39144$,

$\bar{\bar{x}}_5^1 \simeq 3{,}68356$,

$\bar{\bar{x}}_6^1 \simeq 54{,}02514$ und

$\bar{\bar{x}}_7^1 = 0$.

Dieses ist eine zulässige UP-Lösung, jedoch keine UP-Basis-Lösung, da 4 UP-Variablen einen positiven Wert haben und das UP nur 3 Restriktionen hat.

Als zugehöriger Wert der UP-Zielfunktion ergibt sich nach (18) :

$$\begin{aligned}\bar{v}^{UP,1} &\simeq 6 \cdot 5{,}27400 + 4 \cdot 11{,}39144 \\ &\quad + 5 \cdot 3{,}68356 + 6 \cdot 54{,}02514 \\ &= 419{,}77840 .\end{aligned}$$

Das KOP 1 hatte einen Zielfunktionswert von :

$\bar{v}^{KOP\ 1} \simeq 419{,}78481$.

Von Rundungsfehlern abgesehen ist Relation (21) :

$\bar{v}^{UP,1} = \bar{v}^{KOP\ 1}$ also erfüllt.

Aus Behauptung I ergibt sich, dass man implizite zu einer besseren, zulässigen UP-Lösung mit den Vektor-Elementen $\bar{\bar{x}}_j^{T+1}$ $(j=1,\ldots,n)$ kommen kann, wenn es gelingt, in einem weiteren Zyklus T+1 ein KOP T+1 aufzustellen, für welches $\bar{v}^{KOP\ T+1} > \bar{v}^{KOP\ T}$ gilt.

Wie man testen kann, ob ein solches KOP T+1 mit $\bar{v}^{KOP\ T+1} > \bar{v}^{KOP\ T}$ existiert, wird im Abschnitt 2.2.5. beschrieben.

2.2.5 Test auf Verbesserungsfähigkeit des KOP

Wie bereits erwähnt, bedient man sich für diesen Test der optimalen Werte der Dualvariablen des KOP T. Dabei kann man von der Überlegung ausgehen, dass alle für ein KOP überhaupt möglichen Koeffizientenspalten ihrer Entstehung nach Linearkombinationen der n UP-Spalten mit n nicht-negativen Gewichten sein müssen.

Bezeichnet man diese Gewichte ganz allgemein mit g_j $(j=1,\ldots,n)$, so lässt sich jede mögliche KOP-Spalte wie folgt schreiben :

a) $\sum_{j=1}^{n} c_j \cdot \bar{g}_j$ als Zielfunktionswert und

b) $\sum_{j=1}^{n} a_{i,j} \cdot \bar{g}_j \quad (i=1,\ldots,m)$ (22)

als Nebenbedingungskoeffizienten.

Nach dem Preistheorem der linearen Programmierung muss für eine neue KOP-Spalte, die ein besseres KOP T+1 ergeben soll, gelten :

$$\sum_{j=1}^{n} c_j \cdot \bar{g}_j > \sum_{i=1}^{m} \sum_{j=1}^{n} a_{i,j} \cdot \bar{g}_j \cdot \bar{y}_i^{KOP\ T} \qquad (23)$$

Nur, wenn eine neue KOP-Spalte gefunden wird, für die (23) gilt, kann also ein KOP T+1 mit einer besseren Optimal-Lösung existieren.

Es lässt sich nun zeigen, dass sich eine einfachere Bedingung aufstellen lässt, die das gleiche leistet wie (23). Zu diesem Zwecke errechne man zunächst die dem gegenwärtigen Lösungsstande des KOP T entsprechend "aktualisierten" Zielfunktionswerte $\bar{c}_j^{\,T}$ des UP :

$$\bar{c}_j^{\,T} = c_j - \sum_{i=1}^{m} a_{i,j} \cdot \bar{y}_i^{\text{KOP T}} \qquad (24)$$

$$(j=1,\ldots,n).$$

Die $\bar{c}_j^{\,T}$ geben Auskunft darüber, ob eine optimale Gesamtlösung des UP bereits erreicht ist.

Behauptung III :

Nur dann und immer dann, wenn zumindest ein Index j mit $\bar{c}_j^{\,T} > 0$ existiert, lässt sich ein KOP T+1 aufstellen mit $\bar{v}^{\text{KOP T+1}} > \bar{v}^{\text{KOP T}}$, wenn keine Degeneration auftritt.

Beweis :

Angenommen, es existiere ein oder auch mehrere Indices j mit $\bar{c}_j^{\,T} > 0$ oder anders geschrieben :

$$c_j > \sum_{i=1}^{m} a_{i,j} \cdot \bar{y}_i^{\text{KOP T}} \qquad (25)$$

so kann man durch Multiplikation mit einem positiven Gewicht $\bar{g}_j$ erhalten :

$$c_j \cdot \bar{g}_j > \sum_{i=1}^{m} a_{i,j} \cdot \bar{y}_i^{\text{KOP T}} \cdot \bar{g}_j \qquad (26)$$

Bildet man (26) für alle j mit $\bar{c}_j^{\,T} > 0$ mit positivem $\bar{g}_j$

und für alle j mit $\bar{c}_j^T \leq 0$ mit einem $\bar{g}_j = 0$ und summiert über j, so erhält man Bedingung (23) erfüllt. Damit ist der "immer-dann"-Teil der Behauptung bewiesen.

Wenn man andererseits annimmt, es gebe nur noch j mit $\bar{c}_j^T \leq 0$ oder anders geschrieben :

$$c_j \leq \sum_{i=1}^{m} a_{i,j} \cdot \bar{y}_i^{KOP\ T} \quad , \qquad (27)$$

so lassen sich keine nicht-negativen Gewichte $\bar{g}_j$ finden, mit denen man (27) multiplizieren könnte, um dann nach Summation über alle j Bedingung (23) zu erhalten. Damit ist auch der "nur-dann"-Teil der Behauptung bewiesen.

Wenn also Bedingung (27) für alle j erfüllt ist, dann ist das Verfahren beendet, weil sich kein verbessertes KOP T+1 finden lässt.

Existiert zumindest ein j mit $\bar{c}_j^T > 0$, so beginnt ein neuer Zyklus T+1 des Verfahrens mit der neuerlichen Berechnung der TP. Dabei wird immer zumindest eine Linearkombination von UP-Spalten gefunden, die das KOP T+1 gegenüber dem KOP T verbessert (Degeneration ausgeschlossen), da die $\bar{c}_j^T$ dann im System (2) als Zielfunktionskoeffizienten fungieren.

Damit ist auch bewiesen, dass von Fällen der Degeneration abgesehen, stets gilt :

$$\bar{v}^{KOP\ T+1} > \bar{v}^{KOP\ T} .$$

Somit ergibt sich als

Schritt V des Algorithmus :

Man berechne für j=1,...,n Bedingung (24) und prüfe, ob zumindest ein j mit $\bar{c}_j^T > 0$ existiert. Ist das der Fall, so setze man das Verfahren in einem neuen Zyklus mit Schritt I fort.

Andernfalls ist das Verfahren beendet und man errechne

die zugehörige UP-Lösung $\bar{\bar{x}}_1^T$, $\bar{\bar{x}}_2^T$,..., $\bar{\bar{x}}_n^T$ anhand der Gleichungen (14) und (16).

Damit ist das Dekompositionsverfahren vollständig beschrieben.

Betrachten wir wieder das Beispiel. Zunächst sind die $\bar{c}_j^1$ (j=1,...,n) nach Gleichung (24) zu berechnen :

$\bar{c}_1^1 \simeq 5 - (4 \cdot 1{,}38215 + 2 \cdot 0{,}71677) = - 1{,}96214$

$\bar{c}_2^1 \simeq 6 - (3 \cdot 1{,}38215 + 6 \cdot 0{,}71677) = - 2{,}44707$

$\bar{c}_3^1 \simeq 5 - (4 \cdot 1{,}38215 + 0 \cdot 0{,}71677) = - 0{,}52860$

$\bar{c}_4^1 \simeq 4 - (0 \cdot 1{,}38215 + 4 \cdot 0{,}71677) = + 1{,}13292$

$\bar{c}_5^1 \simeq 5 - (6 \cdot 1{,}38215 + 4 \cdot 0{,}71677) = - 6{,}15998$

$\bar{c}_6^1 \simeq 6 - (3 \cdot 1{,}38215 + 2 \cdot 0{,}71677) = + 0{,}42001$

$\bar{c}_7^1 \simeq 8 - (5 \cdot 1{,}38215 + 4 \cdot 0{,}71677) = - 1{,}77783$

Zwei der aktualisierten Zielfunktionswerte des UP sind am Ende des ersten Zyklus noch positiv, nämlich $\bar{c}_4^1$ und $\bar{c}_6^1$. Daher lässt sich noch ein besseres KOP 2 aufstellen.

Der zweite Zyklus des Verfahrens beginnt mit dem nochmaligen Lösen der TP mit den $\bar{c}_j^1$ (j=1,...,n) als Zielfunktionskoeffizienten. Bei TP 1 ist gleich noch vor Beginn der Rechnung wieder eine Hilfszeile wie im ersten Zyklus einzufügen.

Als Lösung der TP im zweiten Zyklus ergibt sich dann :

<u>für TP 1</u> : $\bar{x}_{1,2,1} = \bar{x}_{2,2,1} = \bar{x}_{3,2,1} = 0$;
$\bar{x}_{4,2,1} = 500$ und
$\bar{v}_{2,1}^{TP} \simeq 566{,}4557$ als Zielfunktionswert ;

<u>für TP 2</u> : $\bar{x}_{5,2,2} = \bar{x}_{7,2,2} = 0$;
$\bar{x}_{6,2,2} = 60$ und
$\bar{v}_{2,2}^{TP} \simeq 25{,}2$ als Zielfunktionswert .

Das KOP 2 übernimmt die zwei Variablen aus dem ersten Zyklus und erhält zwei neue dazu. In Tabelle 6 ist es in Tableauform dargestellt :

Tabelle 6 Das KOP 2

BV \ NBV	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	Absolutbeträge
$-v^{KOP\ 2}$	976	372	2000	360	0
u_1	200	200	0	180	200
u_2	-320	180	-2500	180	180
u_3	976	133 1/3	2000	120	200

Seine Primal-Lösung ist :

$\bar{P}^{2}_{1,1} = \bar{P}^{2}_{1,2} = 0$; $\bar{P}^{2}_{2,1} = 1/30$ und $\bar{P}^{2}_{2,2} = 10/9$.

Mit $\bar{P}^{2}_{2,2}$ hat sich hier ein Gewichtungsfaktor für die Lösung des zweiten TP im zweiten Zyklus ergeben, der grösser als 1 ist.

Die Dual-Lösung ist :

$\bar{y}_1^{KOP\ 2} = 1\ 1/3$; $\bar{y}_2^{KOP\ 2} = 0$ und $\bar{y}_3^{KOP\ 2} = 1$.

Der optimale Zielfunktionswert des KOP 2 ist $\bar{v}^{KOP\ 2} = 466\ 2/3$.

Der primalen Lösung des KOP 2 entspricht die UP-Lösung :

$\bar{\bar{x}}_1^{2} = \bar{\bar{x}}_2^{2} = \bar{\bar{x}}_3^{2} = \bar{\bar{x}}_5^{2} = \bar{\bar{x}}_7^{2} = 0$;

$\bar{\bar{x}}_4^{2} = \bar{\bar{x}}^{2}_{4,2,1} = 500 \cdot 1/30 = 16\ 2/3$;

$\bar{\bar{x}}_6^{2} = \bar{\bar{x}}^{2}_{6,2,2} = 60 \cdot 10/9 = 66\ 2/3$

mit einem Zielfunktionswert

$\bar{v}^{UP,2} = 4 \cdot 16\ 2/3 + 6 \cdot 66\ 2/3 = 466\ 2/3.$

Sodann sind die $\bar{c}_j^{\,2}$ (j=1,...,n) nach Gleichung (24) zu berechnen :

$\bar{c}_1^{\,2} = 5 - (4 \cdot 4/3 + 2 \cdot 1) = -2\ 1/3;$

$\bar{c}_2^{\,2} = 6 - (3 \cdot 4/3 + 6 \cdot 1) = -4\ ;$

$\bar{c}_3^{\,2} = 5 - (4 \cdot 4/3 + 0 \cdot 1) = -1/3\ ;$

$\bar{c}_4^{\,2} = 4 - (0 \cdot 4/3 + 4 \cdot 1) = 0\ ;$

$\bar{c}_5^{\,2} = 5 - (6 \cdot 4/3 + 4 \cdot 1) = -7\ ;$

$\bar{c}_6^{\,2} = 6 - (3 \cdot 4/3 + 2 \cdot 1) = 0$ und

$\bar{c}_7^{\,2} = 8 - (5 \cdot 4/3 + 4 \cdot 1) = -2\ 2/3.$

Da kein j mit $\bar{c}_j^{\,2} > 0$ mehr existiert, lässt sich kein besseres KOP 3 mehr aufstellen und das Verfahren ist beendet mit der UP-Lösung :

$\bar{\bar{x}}_1 = \bar{\bar{x}}_2 = \bar{\bar{x}}_3 = \bar{\bar{x}}_5 = \bar{\bar{x}}_7 = 0\ ;$

$\bar{\bar{x}}_4 = 16\ 2/3$ und $\bar{\bar{x}}_6 = 66\ 2/3$ und $\bar{\bar{v}}^{UP} = 466\ 2/3.$

Im folgenden Abschnitt soll nun geprüft werden, ob bei Beendigung des Verfahrens mit $\bar{c}_j^{\,T} < 0$ (j=1,...,n) eine gefundene, zulässige UP-Lösung auch eine optimale UP-Lösung sein muss. Weiterhin soll die Endlichkeit des Verfahrens gezeigt werden.

2.3. Beweis für die Gleichwertigkeit der Optima des UP und des letzten KOP bei normaler Beendigung des Verfahrens und Endlichkeitsbeweis.

2.3.1. Beweis für die Gleichwertigkeit der Optima

Gezeigt wurde oben bereits, dass jeder primalen, zulässigen KOP-Lösung, eine gleichwertige, primale und zulässige UP-Lö-

sung entspricht. Darüber hinaus ist noch zu beweisen

Behauptung IV :

Einer optimalen Primal-Lösung des KOP T mit den Vektor-Elementen $\bar{P}^{T}_{t,k}$ (t=1,...,T; k=1,...,Z) entspricht bei normaler Beendigung des Verfahrens mit $\bar{c}_j^{\,T} \leq 0$ (j=1,...,n) nicht nur irgendeine gleichwertige, zulässige, primale UP-Lösung mit den Vektor-Elementen $\bar{\bar{x}}_j^{\,T}$ (j=1,...,n), sondern diese UP-Lösung ist auch eine optimale UP-Lösung mit $x_j^{*} = \bar{\bar{x}}_j^{\,T}$ für j=1,...,n.

Beweis :

a) Das UP-Dual lautet :

Minimiere

$$\sum_{i=1}^{m} b_i \cdot y_i^{UP} = z^{UP}$$

unter Berücksichtigung der Nebenbedingungen (28)

$$\sum_{i=1}^{m} a_{i,j} \cdot y_i^{UP} \geq c_j \quad \text{für } j = 1,2,\ldots,n$$

$$y_i^{UP} \geq 0 \quad \text{für } i = 1,2,\ldots,m.$$

b) Definition :

Ein KOP heisse vollständig, wenn es für jede mögliche Basislösung eines jeden TP eine entsprechende Variable hat.

Der optimale Zielfunktionswert eines tatsächlich berechneten KOP ist bei Beendigung des Verfahrens gleich dem optimalen Zielfunktionswert des vollständigen KOP. Das ergibt sich daraus, dass

1) Im berechneten KOP stets nur Variable enthalten sein können, die auch im vollständigen KOP enthalten sind, und

2) bei Beendigung des Verfahrens für alle UP-Spalten j gilt $\bar{c}_j^T \leq 0$ $(j=1,\ldots,n)$, woraus sich gemäss dem "nur-dann"-Teil der Behauptung III ergibt, dass das berechnete KOP nicht mehr verbesserungsfähig ist und sich daher im vollständigen KOP keine Variable mehr befinden kann, die man noch mit Vorteil in das berechnete KOP aufnehmen könnte. Daher reichen die im berechneten KOP befindlichen Variablen aus, um ein Optimum des vollständigen KOP zu beschreiben. Die Zielfunktionswerte beider KOP sind dann also gleich.

Wenn G_k die Anzahl der möglichen verschiedenen Basislösungen des k-ten TP ist, dann kann man das vollständige KOP (dessen optimaler Zielfunktionswert ja gleich dem des letzten berechneten KOP ist) wie folgt schreiben :

Maximiere

$$\sum_{k=1}^{Z} \sum_{t=1}^{G_k} \Big(\sum_{j \in N_{TP_k}} c_j \cdot \bar{x}_{j,t,k} \Big) \cdot P_{t,k} = v^{KOP}$$

unter Beachtung der Nebenbedingungen (29)

$$\sum_{k=1}^{Z} \sum_{t=1}^{G_k} \Big(\sum_{j \in N_{TP_k}} a_{i,j} \cdot \bar{x}_{j,t,k} \Big) \cdot P_{t,k} \leq b_i \qquad (i=1,\ldots,m)$$

$$P_{t,k} \geq 0 \qquad (k=1,\ldots,Z; \quad t=1,\ldots,G_k).$$

Das Dual des vollständigen KOP ist dann :

Minimiere

$$\sum_{i=1}^{m} b_i \cdot y_i^{KOP} = z^{KOP}$$

unter Beachtung der Nebenbedingungen (30)

$$\sum_{i=1}^{m} \Big(\sum_{j \in N_{TP_k}} a_{i,j} \cdot \bar{x}_{j,t,k} \Big) \cdot y_i^{KOP} \geq \sum_{j \in N_{TP_k}} c_j \cdot \bar{x}_{j,t,k}$$

$$(k=1,\ldots,Z; \quad t=1,\ldots,G_k)$$

$$y_i^{KOP} \geq 0 \quad (i=1,\ldots,m).$$

c) Das Dekompositionsverfahren ist beendet, wenn gilt :

$$\bar{c}_j^{\,T} = c_j - \sum_{i=1}^{m} a_{i,j} \cdot \bar{y}_i^{KOP\ T} \leq 0 \qquad (31)$$

$$(j=1,\ldots,n)$$

oder umgeformt

$$c_j \leq \sum_{i=1}^{m} a_{i,j} \cdot \bar{y}_i^{KOP\ T} \qquad (32)$$

$$(j=1,\ldots,n).$$

Da die $\bar{y}_i^{KOP\ T}$ $(i=1,\ldots,m)$ in den Relationen (32) alle Restriktionen des UP-Duals (28) erfüllen, geben sie eine zulässige Dual-Lösung des UP an.

Der dieser Lösung zugehörige Zielfunktionswert des UP-Duals ist :

$$\bar{z}^{UP,T} = \sum_{i=1}^{m} b_i \cdot \bar{y}_i^{KOP\ T} \qquad (33)$$

und der des KOP-Duals (die $\bar{y}_i^{KOP\ T}$ sind ja als optimale Duallösung des KOP T entstanden) :

$$\bar{z}^{KOP\ T} = \sum_{i=1}^{m} b_i \cdot \bar{y}_i^{KOP\ T} \quad . \qquad (34)$$

Aus (33) und (34) folgt :

$$\bar{z}^{UP,T} = \bar{z}^{KOP\ T}. \qquad (35)$$

d) Nach dem Dualitätssatz der linearen Programmierung gilt, dass sich im Optimum die Zielfunktionswerte des Primals und des Duals gleichen :

$$\text{Max } v = \text{Min } z. \qquad (36)$$

Da man weiss, dass für das KOP T das Optimum bereits gefunden ist, kann man aus (36) folgern :

$$\bar{v}^{KOP\ T} = \bar{z}^{KOP\ T}. \qquad (37)$$

e) Wie bereits gezeigt, gibt es zu der optimalen Primal-Lösung des KOP T eine zugehörige, zulässige Lösung des UP-Primals mit :

$$\bar{v}^{UP,T} = \bar{v}^{KOP\ T}. \qquad (38)$$

Setzt man (37) in (35) ein, so ergibt sich :

$$\bar{z}^{UP,T} = \bar{v}^{KOP\ T}. \qquad (39)$$

Aus (38) und (39) ergibt sich dann :

$$\bar{v}^{UP,T} = \bar{z}^{UP,T}. \qquad (40)$$

Daraus ist ersichtlich, dass $\bar{v}^{UP}$ seine obere und $\bar{z}^{UP}$ seine untere Grenze erreicht hat. Damit ist auch Behauptung IV bewiesen, dass die UP-Lösung $\bar{\bar{x}}_j^T$ $(j=1,..,n)$ optimal ist, wenn gilt $\bar{\sigma}_j^T \leq 0$ $(j=1,\ldots,n)$.

Es sei hier auch angemerkt, dass dieser Beweis auch für den Fall seine Gültigkeit behält, dass man ausser einem künstlichen TP wie (5) keine weiteren TP berechnet. Denn in diesem Falle ist das vollständige KOP (s. Definition S. 33) mit dem UP identisch und, was zu beweisen ist, schon unmittelbar einsichtig.

2.3.2. Endlichkeitsbeweis

Wie in Abschnitt 2.2.5. bereits gezeigt wurde, verbessert sich der Zielfunktionswert des KOP von Zyklus zu Zyklus, solange noch UP-Spalten mit Index j existieren, für die $\bar{c}_j^T > 0$ gilt, wenn im KOP keine Degeneration auftritt oder dieselbe entsprechend behandelt wird. Damit ist gewährleistet, dass keine optimale KOP-Basislösung nochmals in einem späteren KOP wieder als optimale KOP-Basislösung auftreten kann.

Es gibt eine endliche Anzahl von TP mit einer endlichen Anzahl von TP-Basislösungen. Daher hat das vollständige KOP, das alle nur möglichen KOP-Spalten enthält, eine endliche Anzahl von Spalten.

Die optimalen Basen der berechneten KOP sind immer Kombinationen von Spalten des vollständigen KOP, die sich nur in endlicher Anzahl bilden lassen.

Da keine optimale KOP-Basis nochmals als solche auftreten kann, ist das Verfahren endlich.

Jetzt lässt sich auch Behauptung II, dass aus $\bar{v}^{UP}$ gegen unendlich folgt : $\bar{v}^{KOP\ T}$ geht gegen unendlich, beweisen :

Wenn eine zulässige UP-Lösung ermittelt ist, dann kann das Verfahren nach einer endlichen Zahl von Zyklen nur auf zwei Arten enden :

a) mit einem endlichen Zielfunktionswert des KOP T oder

b) mit einem gegen unendlich gehenden Zielfunktionswert des KOP T.

Angenommen, der Fall a) sei eingetreten. Dann gilt $\bar{c}_j^T \leq 0$ (j=1,...,n) und folglich gemäss Behauptung IV auch, dass $\bar{v}^{KOP\ T}$ dem optimalen Zielfunktionswert des UP gleicht. Das widerspricht aber der Annahme, dass $\bar{v}^{UP}$ gegen unendlich geht. Folglich kann, wenn $\bar{v}^{UP}$ gegen unendlich geht, nie der Fall a) eintreten, sondern immer nur die Alternative b), dass auch $\bar{v}^{KOP\ T}$ gegen unendlich geht.

Damit ist Behauptung II bewiesen.

2.4. Zweistufige Dekomposition

Es ist durchaus möglich, das jeweilige KOP oder die TP auch wieder zu dekomponieren wie zuvor das UP.

Zweifelhaft ist es jedoch, ob das für die TP vom rechnerischen Gesichtspunkt her empfehlenswert ist, da diese i.a. in erster Linie die Funktion haben, Variablen für das KOP zu ermitteln. Das können sie aber schon, wenn die TP nur eine einzige Restriktion haben und ihre Dekomposition sich somit erübrigt. Nur wenn die TP hauptsächlich die Funktion hätten, für das KOP besonders "gute" Variablen zu ermitteln, die besonders schnell zum Ziel führen, könnte es u.U. ratsam sein, jedem Koeffizienten ungleich null aus der UP-Matrix auch zumindest in einem TP erscheinen zu lassen.

Ein KOP hatte bisher immer m Restriktionen. Die Anzahl der eigentlichen Variablen eines KOP T sei mit q_T bezeichnet. Wie bereits gezeigt wurde, kann q_T nie grösser als m+Z werden. Eine Dekomposition eines primalen KOP erscheint nicht sinnvoll, weil man damit weder die Anzahl der Restriktionen überhaupt noch die Anzahl der maximal möglichen eigentlichen KOP-Variablen wesentlich verringern kann.

Hingegen ist es möglich, dass die Dekomposition des Duals eines KOP T nach diesem Verfahren Vorteile hat, insbesondere, wenn das primale KOP T sehr viel mehr Restriktionen als eigentliche Variablen hat. Zu diesem Zwecke bilde man das duale KOP T :

Minimiere

$$\sum_{i=1}^{m} b_i \cdot y_i^{\text{KOP T}} = z^{\text{KOP T}}$$

unter Beachtung der Nebenbedingungen (41)

$$\sum_{i=1}^{m} \Big(\sum_{j \, N_{TP_k}} a_{i,j} \cdot \bar{x}_{j,t,k} \Big) \cdot y_i^{\text{KOP T}} \geq \sum_{j \, N_{TP_k}} c_j \cdot \bar{x}_{j,t,k}$$

$(k=1,\ldots,Z) \quad (t=1,..,T)$

$y_i^{KOP\ T} \geq 0 \quad (i=1,\ldots,m).$

Wenn das primale KOP T q_T Variablen hatte , dann hat das duale KOP T q_T Restriktionen. Auch jedes KOP der zweiten Dekompositionsstufe hat dann q_T Restriktionen.

Gilt $q_T << m$, so kann die Anzahl der in einem beliebigen KOP der zweiten Stufe höchstens zu berechnenden eigentlichen Variablen in jedem Falle bis auf nahezu q_T verkleinert werden. Das entspricht einer Reduktion der Anzahl der Restriktionen des primalen KOP T der ersten Stufen und kann zumindest unter dem Gesichtspunkt der Rechengenauigkeit - wenn nicht auch unter dem der benötigten Rechenzeit - von Vorteil sein.

Das duale KOP T ist dann wie ein UP zu behandeln. Allerdings hat dieses Verfahren einen Nachteil : Wenn für das UP die Nullpunktlösung zulässig war, dann haben das duale KOP T ebenso wie die daraus resultierenden KOP der 2. Stufe keine zulässige Nullpunktlösung, so dass dann eine erste Phase des Verfahrens notwendig ist, um eine zulässige Ausgangslösung zu finden. Das bedingt jedoch einen relativ grossen Aufwand an Rechenzeit. Hat man eine zulässige Ausgangslösung für (41) gefunden, dann sucht man in einer zweiten Phase eine optimale Lösung. Stellt man dagegen fest, dass eine zulässige Primal-Lösung für (41) nicht existiert, so folgt gemäss dem Dualitätssatz der linearen Programmierung, dass der Zielfunktionswert des primalen KOP T gegen unendlich geht. Geht andererseits der Zielfunktionswert des dualen KOP T gegen unendlich - der Fall könnte nur vorkommen, wenn die Nullpunktlösung des KOP T nicht zulässig wäre - , dann hat das KOP T nach dem Dualitätssatz keine zulässige, primale Lösung.

Selbstverständlich gibt es ein direkt auf das KOP T anwendbares Verfahren, welches der Dualbildung des KOP T und der Anwendung des vorliegenden Algorithmus darauf entspricht. Diese Frage soll hier jedoch nicht mehr untersucht werden.

3. Erste praktische Tests des Algorithmus auf elektronischen Datenverarbeitungs-Anlagen

3.1. Das Testprogramm

Der Verfasser hat in der Programmiersprache ALGOL ein Testprogramm für diesen Dekompositions-Algorithmus geschrieben. Dieses Testprogramm benutzt für Daten nur den Kernspeicher der Rechenanlage, benutzt also keine externe Datenspeicherung. Das Programm ist so umfangreich, dass es auf einer CD 3300, die für ALGOL-Programme 32K Kernspeicher verfügbar hat, praktisch nur in segmentierter Form ausgeführt werden konnte, d.h., dass ein Teil des Programmes während der Rechnung auf eine Platte ausgelagert wird und bei Bedarf im Austausch gegen andere Programmteile in den Kernspeicher geholt wird.

Das Programm lässt nur einstufige Dekomposition zu. Eine Dekomposition des KOP ist damit also nicht möglich. Künstliche TP ähnlich wie (5) werden programm-intern definiert und mitberechnet, jedoch nicht nur eines, sondern mehrere, da die für ein TP höchstzulässige Anzahl von Variablen im Programm begrenzt wurde, um den für die Speicherung der TP-Lösungen benötigten Speicherplatz zu reduzieren.

Die TP und das KOP werden in jedem Zyklus wieder von Grund auf optimiert vermittels einer vom Verfasser selbst programmierten Prozedur, die das normale Simplex-Verfahren beinhaltet.

Das Programm kann grundsätzlich alle linearen Optimierungsprobleme lösen. Ausgiebig getestet worden ist es jedoch nur für solche Probleme, bei denen die Nullpunktlösung zulässig ist.

3.2. Die Rechentests

Die Rechentests wurden durchgeführt, um nähere Informationen über den Rechenzeitbedarf, die Rechengenauigkeit und den

Speicherplatzbedarf zu erhalten.

Die folgenden Angaben über die jeweils benötigte Rechenzeit haben aus mehreren Gründen jedoch nur eine beschränkte Aussagekraft und können Grössenordnungen und Grössenverhältnisse nur andeuten :

(1) Es war dem Verfasser nicht möglich, die Zeit für die Abarbeitung bestimmter Programmteile zu messen, sondern die angegebene Zeit ist jeweils die gesamte JOB-Zeit, wenn nicht ausdrücklich etwas anderes vermerkt ist. Darin sind also z.B. enthalten : die Zeit für die Übersetzung des Programms, die Zeit für die Ein- und Ausgabe der Daten und die Zeit, die der Computer für die Organisation des Auswechselns der Programmsegmente im Kernspeicher benötigt. Die erste Zeit ist bezogen auf die gespeicherte Datenmenge fix, die zweite nimmt etwa linear und die dritte sehr stark progressiv zu.

(2) Programm-technisch gesehen, ist das Testprogramm vermutlich recht ungeschickt und damit zeitaufwendig aufgebaut, da es das erste Programm war, welches der Verfasser überhaupt geschrieben hat. Darüberhinaus lässt der Algorithmus dem Programmierer viel Freiheit, das Programm zu organisieren und damit die Rechenzeit zu beeinflussen. Wichtig ist in dieser Hinsicht z.B., wie man die TP und das KOP löst, ob und gegebenenfalls welche KOP-Variablen man wieder streicht und ob man die KOP-Variablen von Zyklus zu Zyklus speichert oder aus den TP-Lösungen wieder neu errechnet, wie es im Testprogramm geschieht.

Die ersten Rechentests mit Problemen mit höchstens 700 Matrix-Koeffizienten wurden auf einer Elektrologica X1 ausgeführt, bei der alle output-Befehle ohne zeitliche Pufferung stets sofort noch während des Programmlaufs über den direkt angeschlossenen Schnelldrucker ausgegeben wurden. Durch Beobachtung der laufenden Schnelldrucker-Ausgabe konnte man dabei schon feststellen, dass die Berechnung der TP i.a. den grössten Teil der Programmlaufzeit beanspruchte. Man könnte

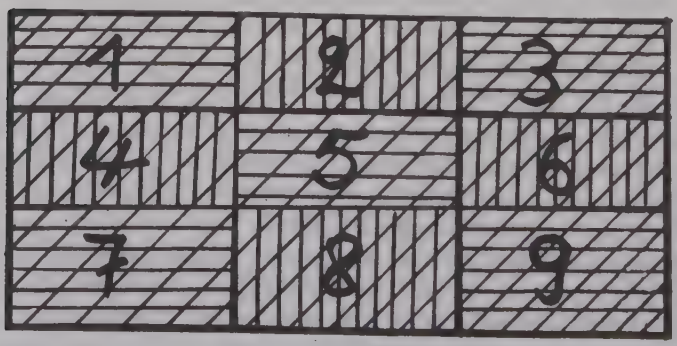
daher meinen, es sei empfehlenswert, den Zeitaufwand zur Ermittlung der KOP-Variablen zu minimieren und daher möglichst wenige (1) und möglichst kleine TP (1 Restriktion) einzuteilen. Dadurch erhöht man aber die Zahl der notwendigen Zyklen, weil dadurch jedesmal weniger neue Variable in das KOP kommen, als sonst und weil möglicherweise TP mit mehr Restriktionen bessere KOP-Variable liefern als TP mit nur einer Restriktion. Hier gilt es also ein Optimum zu finden, denn durch die Einteilung der TP wird die notwendige Rechenzeit erheblich beeinflusst.

Im folgenden seien einige Beispiele gezeigt, wie man unstrukturierte oder auch strukturierte Matrizen einteilen kann.

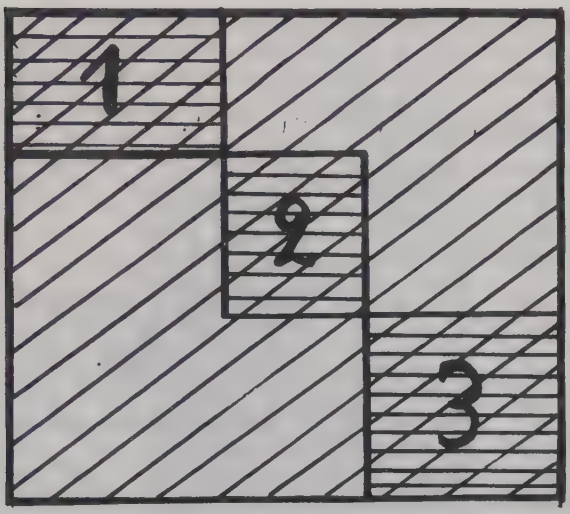
Die Darstellung wird schematisch skizziert :

Die Matrix des UP wird als Rechteck dargestellt und ihre mit von null verschiedenen Koeffizienten besetzten Teile werden schräg schraffiert. Die nicht unbedingt zweckmässig aber auch nicht falsch eingeteilten TP werden als waagerecht oder senkrecht schraffierte Rechtecke eingezeichnet und durchnummeriert.

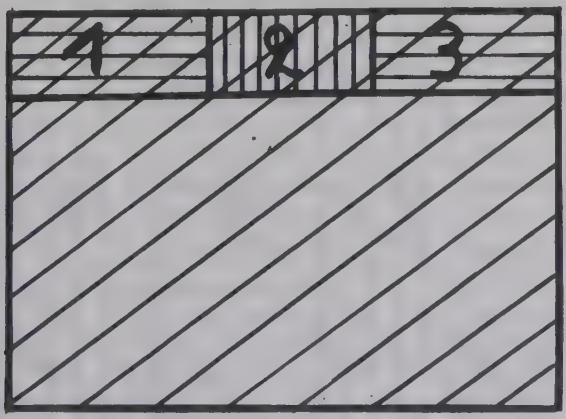
a) Hier ist jeder Koeffizient der UP-Matrix in genau einem TP enthalten. Jeder Index j und i des UP taucht dagegen in drei verschiedenen TP auf.



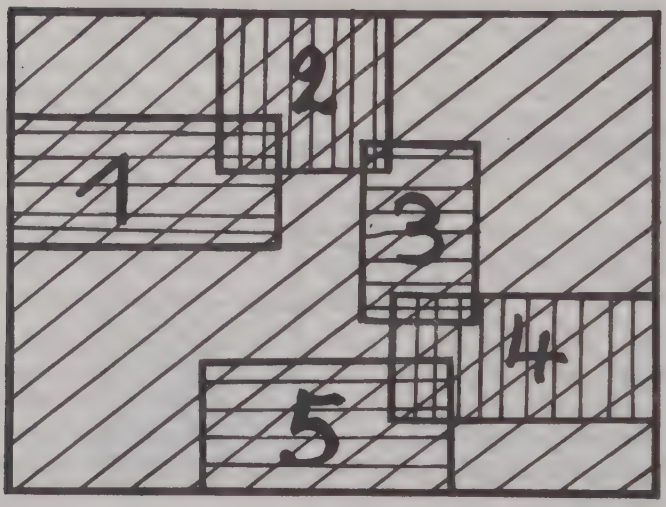
b) Bei dieser Einteilung ist jeder Index j und jeder Index i des UP in genau einem TP enthalten.



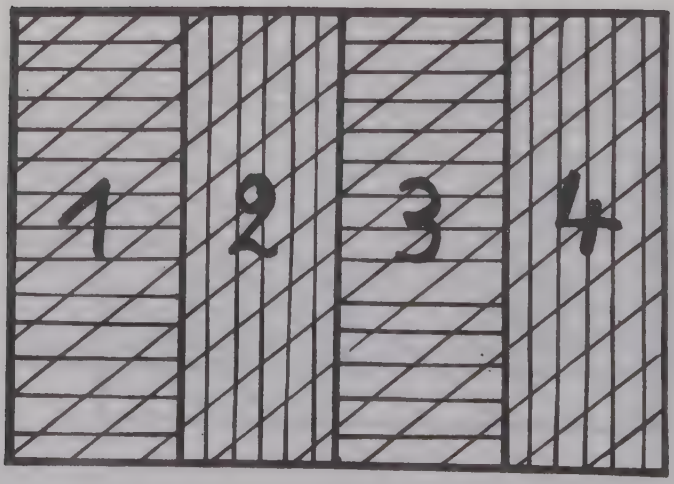
c) Hier ist zwar jeder Index j, nicht jedoch jeder Index i des UP in einem TP enthalten.



d) In diesem Falle sind Koeffizienten des UP in mehreren TP enthalten.



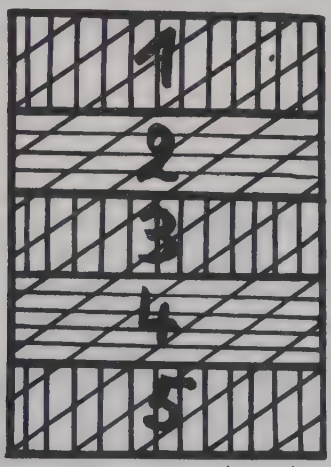
e)



Diese Einteilung hat einen Vorteil :

Wenn nämlich ein TP hier keine TP-zulässige, duale Lösung besitzt (der Zielfunktionswert des primalen TP gegen unendlich geht), dann geht auch der Zielfunktionswert des UP gegen unendlich; denn, da die zulässige duale Lösungsmenge des UP der Durchschnitt der zulässigen dualen Lösungsmengen aller TP ist, gibt es dann keine zulässige duale UP-Lösung.

f)



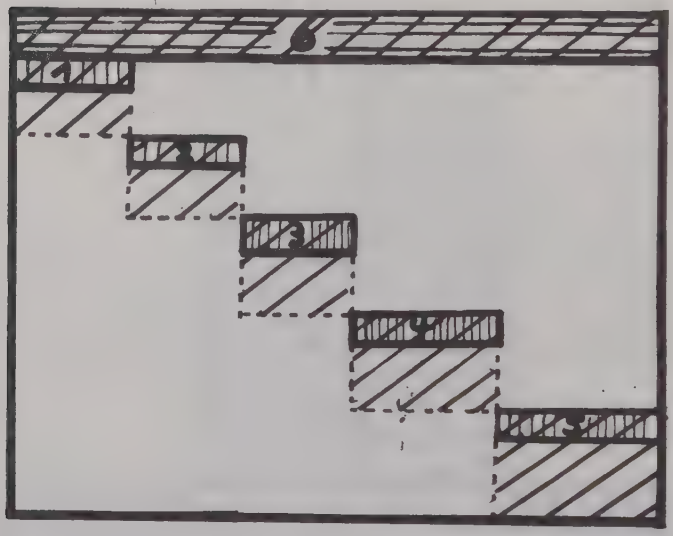
Diese TP-Einteilung kann der Früherkennung der Tatsache dienen, dass für das UP keine zulässige, primale Lösung existiert. Wenn nämlich ein TP keine TP-zulässige, primale Lösung hat, dann hat auch das UP keine zulässige, primale Lösung; denn die zulässige Lösungsmenge des UP ergibt sich als Durchschnitt der zulässigen Lösungsmengen aller TP.

Um das festzustellen, könnte man bei Bedarf die TP wie ein UP nochmals dekomponieren.

Ob diese TP-Dekompesition ratsam ist, ist allerdings zweifelhaft, da die Tatsache, dass alle TP primal zulässige Lö-

sungen haben, keineswegs auf die Existenz einer primal zulässigen Lösung des UP schliessen lässt.

g) Ein solches UP könnte man vermutlich besser mit dem Dekompositionsverfahren von Dantzig und Wolfe lösen, weil man damit auch gleich die Anzahl der Restriktionen vermindern könnte, obwohl die gezeigte TP-Einteilung zu kleineren TP führt als bei Dantzig.



Alle gezeigten TP-Einteilungen sind zulässig in dem Sinne, dass sie das Dekompositions-Verfahren von Rundungsfehlern abgesehen zu einem gleich guten Ergebnis kommen lassen wie die auf das gesamte UP angewendete Simplex-Methode.

Die TP dürfen teilweise sich selbst oder auch nicht mit von null verschiedenen Koeffizienten besetzte Matrix-Flächen überdecken. Andererseits brauchen sie nicht alle von null verschiedenen Koeffizienten zu umfassen.

Besondere Matrix-Strukturen können ausgenutzt werden, wenn das auch nicht wie bei dem Dekompositions-Verfahren von

Dantzig etwa zu einer Reduktion der Restriktionen-Anzahl führt.

Im folgenden seien nun einige Test beschrieben. Die Beispiele wurden grundsätzlich einmal mit der im Testprogramm enthaltenen normalen Simplex-Methode gelöst und i.a. mehrere Male auf verschiedene Art und Weise mit dem Dekompositions-Algorithmus. Die mit der normalen Simplex-Methode erzielten Ergebnisse dienen als Vergleichsgrundlage für die mit dem Dekompositions-Verfahren erzielten Ergebnisse. Sollte der Verfasser die normale Simplex-Methode ineffektiv programmiert haben in Bezug auf die benötigte Rechenzeit, so würde dieser Nachteil vermutlich ebensosehr für die Dekompositions-Ergebnisse gelten, da in jedem Zyklus das KOP und die TP mit dieser Simplex-Prozedur von Grund auf neu berechnet wurden.

Durch die Tatsache, dass nahezu alle Rechenläufe nur in segmentierter Form ausgeführt werden konnten, wurden die Laufzeiten der Dekompositions-Rechnungen systematisch mehr verlängert als die vergleichbaren Ergebnisse der normalen Simplex-Methode. Denn da das Dekompositions-Verfahren mehr Daten speichert als die normale Simplex-Methode, können weniger Programmteile im Kernspeicher gespeichert werden. Ausserdem benutzt die normale Simplex-Methode nur einen kleinen Teil des gesamten Programms. Aus beiden Gründen müssen beim Dekompositions-Verfahren die Programm-Segmente häufiger als bei der normalen Simplex-Methode im Kernspeicher ausgewechselt werden. Der für die Organisation dieses Austausches notwendige Zeitaufwand ist in den angegebenen JOB-Zeiten mit enthalten.

Jedes Testbeispiel wurde i.a. mehrmals mit dem Dekompositions-Verfahren gelöst. Bei diesen Rechnungen wurde die Anzahl der TP und auch deren Grösse variiert, um zu untersuchen, wie dadurch die Ergebnisse beeinflusst werden. Die Einteilung der TP wurde stets so vorgenommen, dass jede Variable in genau einem der eingeteilten TP enthalten war. Für jedes eingeteilte TP wurde weiterhin programm-intern ein künstliches TP ähnlich wie (5) sicherheitshalber mitgerechnet.

Die künstlichen TP umfassten jeweils die gleichen Variablen wie die zugehörigen, extern eingeteilten TP.

3.2.1. Die Testbeispiele

In diesem Abschnitt sollen die berechneten Testbeispiele kurz charakterisiert werden. Jedes Testbeispiel erhält eine dreiteilige Kenn-Nr., aus der schon wichtige Eigenschaften desselben hervorgehen. So bedeutet z.B. die Nummer B 40-100-50, dass es sich um ein Beispiel mit 40 Nebenbedingungen, 100 Variablen und einer Koeffizientenmatrix handelt, die zu 50 % mit von Null verschiedenen Koeffizienten besetzt ist.

a) Testbeispiel B 60-60-50 :

- Maximierungsproblem;
- 60 Nebenbedingungen (m = 60);
- 60 Variablen (n = 60);
- Nullpunktlösung zulässig;
- 50 % aller Matrixkoeffizienten positiv, der Rest ist gleich null;
- die Nullen sind wie zufällig über die Matrix verteilt;
- die positiven Koeffizienten liegen zwischen 1 und 10 und sind zufallsähnlich ermittelt.

b) Testbeispiel B 40-100-50 :

- 40 Nebenbedingungen (m = 40);
- 100 Variablen (n = 100);
- sonst wie B 60-60-50.

c) Testbeispiel B 100-50-100 :

- Minimierungsproblem;
- 100 Nebenbedingungen (m = 100);
- 50 Variablen (n = 50);
- Matrix zu 100 % mit zufallsähnlich ausgewählten Koeffizienten zwischen 1 und 10 besetzt;
- Nullpunktlösung nicht zulässig, da alle Restriktionen von folgendem Typ sind : $b_i \leq \sum_{j=1}^{n} a_{ij} \cdot x_j$

d) Testbeispiel B 40-100-100 :

- 40 Nebenbedingungen (m = 40);
- 100 Variablen (n = 100);
- sonst wie B 100-50-100.

e) Testbeispiel B 80-500-50 :

- 80 Nebenbedingungen (m = 80);
- 500 Variablen (n = 500);
- sonst wie B 40-100-50.

f) Testbeispiel B 100-500-10 :

- 100 Nebenbedingungen (m = 100);
- 500 Variablen (n = 500);
- 10 Restriktionen, die alle Variablen umfassen;
- 9 diagonal über die Matrix verteilte Koeffizientenblöcke, die jeweils 10 Nebenbedingungen und ca. 55 Variablen umfassen;
- alle übrigen Koeffizienten gleich null;
- Matrix zu 10 % mit von null verschiedenen Koeffizienten besetzt;
- sonst wie B 60-60-50.

3.2.2. Die Rechenzeiten

Wenn nicht ausdrücklich etwas anderes vermerkt ist, gilt als Rechenzeit die Zeit, während der der Programmlauf von der Zentraleinheit des Computers tatsächlich bearbeitet wird.

a) Testbeispiel B 60-60-50 :

- Benutzte Rechenanlage : CD 3300.
- Rechenzeit mit der einfachen Simplex-Methode : 3 Minuten 46 Sekunden.

Aus der Tabelle 6 können die Zeiten für einige Läufe mit dem Dekompositions-Verfahren entnommen werden, bei denen eine unterschiedliche Anzahl verschieden grosser TP eingeteilt

wurde. In der waagerechten Kopfzeile ist angegeben, wieviel Nebenbedingungen die eingeteilten TP hatten. In den senkrechten Eingangsspalten ist angegeben, wieviele TP jeweils eingeteilt wurden und wieviele Variablen die TP i.a. hatten. Zu der angegebenen Anzahl der eingeteilten TP hinzuzurechnen ist jeweils eine gleichgrosse Anzahl programm-intern definierter, künstlicher TP, die stets mitberechnet wurden.

Tabelle 6 : Rechenzeiten für Beispiel B 60-60-50 für Dekompositions-Läufe in Minuten und Sekunden

Anzahl der Variablen je TP	Anzahl der TP *	Anzahl der NB je TP		
		60	30	6
30	2	18/12	12/17	8/45
20	3	16/44	9/16	7/01
15	4	10/45	10/17	7/37
10	6	11/35	9/11	9/30
6	10	9/06	6/38	8/55

* : ohne künstliche TP
NB : Nebenbedingungen

Diese Tabelle vermittelt den Eindruck, dass mit abnehmender Anzahl von Nebenbedingungen in den TP der Rechenzeitbedarf abnimmt. Ebenso scheint der Rechenzeitbedarf mit steigender Anzahl der TP abzunehmen. Wie weitere Tests zeigen werden, ist letzteres jedoch nur bis zu einer gewissen Grenze der Fall.

Eine weitere Serie von Dekompositions-Testläufen mit dem Beispiel B 60-60-50 wurde mit ausschliesslich künstlichen TP durchgeführt. Tabelle 7 zeigt die Resultate dieser Rechnungen.

Tabelle 7 : JOB-Zeiten und Anzahl der Verfahrenszyklen für Testbeispiel B 60-60-50 ausschliesslich künstlichen TP

Anzahl künstlicher TP	Anzahl der Verfahrens-Zyklen	Rechenzeiten [min / sek]
2	15	7/30
3	12	5/38
4	10	4/50
5	6	3/31
6	7	4/24
10	7	5/34

Hier sind die Zeiten eng mit der benötigten Anzahl von Verfahrens-Zyklen korreliert. Diese JOB-Zeiten sind fast durchweg kürzer als die unter Verwendung der nicht-künstlichen TP ermittelten Zeiten. Nur ein Rechenlauf war hier schneller als die Vergleichszeit von 3 min 46 sek für die normale Simplex-Methode.

b) Testbeispiel B 40-100-50 :

Das Beispiel wurde wieder auf der CD 3300 mit segmentierten Programmläufen berechnet.

Mit der normalen Simplex-Methode ergab sich eine JOB-Zeit von 6 min 04 sek.

Die JOB-Zeiten für Dekompositions-Testläufe bei Einteilung einer unterschiedlichen Anzahl verschieden grosser TP können aus der Tabelle 8 entnommen werden.

Tabelle 8 : JOB-Zeiten für Testbeispiel B 40-100-50
[min, sek]
Vergleichszeit : 6/04 [min, sek]

Anzahl der Variablen je TP	Anzahl der TP (ohne künstl. TP)	Anzahl der NB je TP		
		40	20	5
34	3	12/23	10/24	8/29
25	4	14/07	9/22	11/52
20	5	14/12	8/41	7/35
17	6	12/40	8/33	6/41
15	7	12/17	8/23	7/08
13	8	15/33	12/34	8/05
10	10	8/03	7/01	8/37
8	13	7/41	7/12	4/43
7	15	9/55	9/44	6/21
6	17	13/04	9/13	7/17

Diese Tabelle bestätigt, dass mit abnehmender Anzahl von Nebenbedingungen in den TP der Rechenzeitbedarf i.a. abnimmt. Tendenziell nimmt auch die Anzahl der benötigten Verfahrens-Zyklen mit der Anzahl der Nebenbedingungen in den TP ab, wie Tabelle 9 zeigt.

Weiter zeigt Tabelle 8, dass mit zunehmender Anzahl der TP der Rechenzeitbedarf tendenziell zunächst fällt und dann wieder steigt, wobei die optimale Anzahl von TP für dieses Beispiel ungefähr bei 13 liegt.

Tabelle 9 : Anzahl der benötigten Verfahrenszyklen für Testbeispiel B 40-100-50

Anzahl der Variablen je TP	Anzahl der TP (ohne künstl. TP)	Anzahl der NB je TP		
		40	20	5
34	3	16	16	16
25	4	19	13	16
20	5	12	9	10
17	6	12	8	8
15	7	11	8	9
13	8	14	10	8
10	10	7	7	7
8	13	6	7	6
7	15	7	9	6
6	17	11	7	7

In Tabelle 8 gibt es Ausreisser-Werte, die den aufgezeigten Tendenzen widersprechen. Bei zeilenweiser Betrachtung sind dies die Werte 11/52 und 8/37, bei spaltenweiser Betrachtung insbesondere die Werte 12/23, 15/33, 12/34, 9/44 und 11/52. Diese Unregelmässigkeiten liegen z.T. darin begründet, dass das Verfahren aus irgendeinem Grunde in diesen Fällen relativ viele oder wenige Zyklen brauchte, wie Tabelle 9 zeigt. Die benötigten Rechenzeiten hängen also auch von der Anzahl der benötigten Zyklen ab. Das wird insbesondere aus Tabelle 10 deutlich, in der Rechenzeiten und Zyklen-Anzahl bei der Berechnung ausschliesslich künstlicher TP gegenübergestellt sind.

Tabelle 10 : JOB-Zeiten und Anzahl der Verfahrenszyklen und maximale Anzahl der KOP-Variablen für Testbeispiel B 40-100-50 mit ausschliesslich künstlichen TP

Vergleichszeit : 6/04 [min, sek]

Anzahl künstl. TP	Anzahl der TP-Variablen	Anzahl der Zyklen	JOB-Bearbeitungszeiten [min, sek]	maximale Anzahl der KOP-Variablen
3	34	16	5/48	22
4	25	12	5/10	23
5	20	11	4/46	22
6	17	12	5/21	26
7	15	9	4/42	24
8	13	9	4/35	25
10	10	8	4/21	24
13	8	7	4/13	23
15	7	6	4/00	23
17	6	6	3/43	23
20	5	7	4/39	26
25	4	4	3/56	29
34	3	6	4/38	34
50	2	4	4/24	50

Mit steigender Anzahl der TP wird die Anzahl der benötigten Zyklen tendenziell immer kleiner, während die Rechenzeiten schliesslich wieder anzusteigen scheinen. Dieser schliessliche Wiederanstieg findet seine Begründung in dem zuletzt starken Ansteigen der maximalen Anzahl der KOP-Variablen.

In Tabelle 9 lag nur eine JOB-Zeit unter der Vergleichszeit. In Tabelle 10 liegen alle darunter. Die Vergleichszeit von 6/04 liegt um nahezu 60 % über der besten JOB-Zeit aus Tabelle 10.

Offensichtlich lohnt es sich unter zeitlichen Gesichtspunkten, das Verfahren nur mit künstlichen TP zu rechnen. Zumindest ist das der Fall, wenn die Koeffizienten-Matrix so gleichförmig wie bei diesen Testbeispielen aussieht.

Auch wird die Verwandtschaft des Verfahrens mit der revidierten Simplex-Methode besonders deutlich, wenn man nur mit künstlichen TP rechnet; diese leisten dann nämlich nichts anderes, als aus einem bestimmten Bereich von Variablen immer diejenige mit dem maximalen Zielfunktionskoeffizienten für das variable Rechentableau des KOP auszuwählen.

c) Testbeispiel B 100-50-100 :

Die Rechnungen erfolgten wieder auf der CD 3300 mit segmentiertem Programm. Für die simultane Rechnung mit der normalen Simplex-Methode wurde das Zwei-Phasen-Verfahren benutzt. Zur Errechnung einer zulässigen Lösung musste 616 mal ein Basistausch vorgenommen werden. In Phase II gab es dann nur noch 69 Iterationen. Entsprechend lang war die Rechenzeit mit 39 min 29 sek.

Aus Tabelle 11 gehen die JOB-Zeiten für einige Dekompositions-Läufe mit ausschliesslich künstlichen TP hervor.

Tabelle 11 : JOB-Zeiten und Anzahl der Verfahrens-Zyklen für Testbeispiel B 100-50-100 mit ausschliesslich künstlichen TP

Vergleichszeit : 39/29 [min, sek]

Anzahl der TP	Anzahl der TP-Variablen	JOB-Bearbeitungszeiten [min, sek]	Anzahl der Verfahrens-Zyklen
2	25	44/04	15
3	17	27/58	9
5	10	26/03	7
7	8	27/58	6
10	5	26/46	6
13	4	21/56	4
17	3	39/20	7 (5)
25	2	32/12	4

Die JOB-Zeiten sind wieder mit der Anzahl der durchlaufenen Verfahrens-Zyklen korreliert. Von dem Beispiel mit 2 TP abgesehen liegen alle Zeiten unter der Vergleichszeit. Bei der Rechnung mit 17 TP wurden nach Erreichen des Optimums noch zwei völlig überflüssige Zyklen und ein nahezu überflüssiger Zyklus, der nur noch den zwanzigmillionsten Teil des optimalen Zielfunktionswertes brachte, durchgerechnet, weil eine programminterne Prüfnull zu klein gewählt worden war.

Die Vergleichszeit liegt um etwa 80 % über der Bestzeit der Tabelle 11. Diese Ergebnisse konnten bei einer Matrix mit erheblich mehr Zeilen als Spalten trotz nur einstufiger

Dekomposition wohl nur deshalb relativ günstig ausfallen, weil die normale Simplex-Methode allein in Phase I schon über 600 Iterationen machen musste.

Besonders interessant an diesem Beispiel ist die Tatsache, dass es doppelt soviele Nebenbedingungen wie Variablen hat und dass sich in seiner Koeffizientenmatrix keine Nullen befinden. Da trotzdem im Vergleich zur normalen Simplex-Methode bei den Dekompositionsläufen so günstige Rechenzeiten erzielt wurden, wird klar, dass die Vorteilhaftigkeit dieses Verfahrens nicht wie bei der revidierten Simplex-Methode darauf beruht, dass die Matrix viele Nullen enthält und erheblich mehr Variablen hat als Restriktionen.

d) Testbeispiel B 40-100-100 :

Die normale Simplex-Methode benötigte 302 Iterationen und eine JOB-Bearbeitungszeit von 15 Minuten und 14 Sekunden auf der CD 3300 mit segmentiertem Rechenlauf.

Die JOB-Bearbeitungszeiten für einige Dekompositions-Läufe mit ausschliesslich künstlichen TP gehen aus Tabelle 12 hervor.

Tabelle 12 : JOB-Zeiten und Anzahl der Verfahrens-Zyklen für Testbeispiel B 40-100-100 mit ausschliesslich künstlichen TP

Vergleichszeit : 15/14 [min, sek]

Anzahl der TP	Anzahl der TP-Variablen	Anzahl der Verfahrens-Zyklen	JOB-Bearbeitungszeiten [min, sek]	maximale Anzahl von KOP-Variablen
50	2	4	9/45	50
34	3	3	7/52	34
25	4	6	10/03	31

Fortsetzung Tabelle 12

Anzahl der TP	Anzahl der TP-Variablen	Anzahl der Verfahrens-Zyklen	JOB-Bearbeitungszeiten [min, sek]	maximale Anzahl von KOP-Variablen
20	5	5	7/13	20
13	8	4	6/14	21
10	10	4	5/31	16
8	13	7	6/14	14
6	17	9	8/23	15
4	25	11	8/13	14
3	34	14	10/25	14
2	50	18	11/22	13

Alle JOB-Zeiten liegen unter der Vergleichszeit, die beste sogar um über 63 %. Die Rechnung mit 25 TP hat unverhältnismässig viele Verfahrens-Zyklen benötigt und eine entsprechend erhöhte JOB-Zeit. Im übrigen fallen die JOB-Zeiten mit fallender TP-Anzahl zunächst, weil die maximal in einem KOP zu berechnende Anzahl von KOP-Variablen sinkt, steigt aber dann wieder an, weil diese Wirkung durch die steigende Anzahl von notwendigen Verfahrens-Zyklen überkompensiert wird.

e) Testbeispiel B 80-500-50 :

Dieses Beispiel wurde auf einer AEG-Telefunken TR 440 gerechnet, die über einen genügend grossen Kernspeicher verfügt, um alle Daten und das Programm gleichzeitig darin unterbringen zu können, und die um ein Vielfaches

schneller rechnet als die CD 3300. Bei den für diese Rechenanlage angegebenen Zeiten handelt es sich ausnahmsweise um reine Rechenzeiten, die nicht die Zeit für die Übersetzung des Programms in die Maschinensprache enthält.

Die reine Rechenzeit für dieses Beispiel mit der normalen Simplex-Methode betrug :

6 Minuten 27 Sekunden.

Zwei Dekompositionsläufe mit ausschliesslich künstlichen TP wurden durchgeführt. Die Rechenzeiten betrugen für den Lauf mit

- 20 TP zu je 25 Variablen : 3 min 39 sek und
- 25 TP zu je 20 Variablen : 3 min 09 sek. .

Die Ergebnisse in Tabelle 10 (S. 54) legen die Vermutung nahe, dass man für 500 Variablen mehr als 25 TP einteilen müsste, um noch kürzere Rechenzeiten zu erzielen. Der Rechenlauf mit 25 TP realisierte jedoch immerhin schon eine Rechenzeiteinsparung von über 50 %. Dabei war der letzte Zyklus in diesem Rechenlauf noch völlig überflüssig, da eine programminterne Prüfnull zu klein gewählt worden war, mit deren Hilfe festgestellt werden sollte, ob es noch positive Zielfunktionskoeffizienten gibt. Das ist nicht unerheblich, da für den genannten Rechenlauf nur 10 Zyklen durchlaufen wurden.

f) Testbeispiel B 100-500-10 :

Auch dieses Beispiel wurde auf einer AEG-Telefunken TR 440 gerechnet. Mit der normalen Simplex-Methode ergab sich eine Rechenzeit von :

6 Minuten 13 Sekunden.

Obwohl dieses Beispiel 20 Nebenbedingungen mehr hat als B 80-500-50 ist die Rechenzeit dennoch geringer als bei jenem. Das ist wahrscheinlich eine Folge der Tatsache, dass die Koeffizientenmatrix von B 100-500-10 mit 10 % viel

geringer mit von null verschiedenen Koeffizienten besetzt ist als es bei B 80-500-50 mit einer Besetzung von 50 % der Fall ist.

Zwei Dekompositionsläufe mit ausschliesslich künstlichen TP wurden durchgeführt. Die Rechenzeiten betrugen für den Lauf mit

- 25 TP zu je 20 Variablen : 2 min 16 sek und
- 42 TP zu je ca. 12 Variablen : 2 min 04 sek. .

Das entspricht einer Rechenzeiteinsparung von ca. 63 und 67 %. Als beachtlich erscheint, dass trotz erheblicher Veränderung der Anzahl der eingeteilten TP die erzielten Rechenzeiten in beiden Fällen mit weniger als 40 % der Vergleichszeit relativ stabil sind. Das gibt zu der Erwartung Anlass, dass man mit einiger Erfahrung für LP-Probleme TP-Einteilungen treffen kann, von denen man sagen kann, dass sie mit einer grossen Wahrscheinlichkeit zu bestimmten Rechenzeiteinsparungen führen.

In Tabelle 13 ist dargestellt, welche Rechenzeiteinsparungen bei den einzelnen Beispielen bestenfalls erzielt werden konnten.

Tabelle 13 : Übersicht über Rechenzeiten verschiedener Beispiele in Minuten und Sekunden

(1) Beispiel	(2) Nullpunktlösung zulässig	(3) Rechenzeit normale Simplex-Methode	(4) beste Rechenzeit Dekompositions-verfahren	(5) Rechenzeiteinsparung $(5)=100-\frac{(4)}{(3)}\cdot 100$
60- 60- 50	JA	3/46	3/31	6
40-100- 50	JA	6/04	3/43	39
100- 50-100	NEIN	39/29	21/56	44
40-100-100	NEIN	15/14	5/31	64
80-500- 50	JA	6/27 *	3/09 *	51
100-500- 10	JA	6/13 *	2/04 *	67

* Bei diesen Angaben handelt es sich um reine Rechenzeiten, die auf einer TR 440-Anlage erzielt wurden.

Man fragt sich angesichts dieser Zusammenstellung, warum bei dem Beispiel B 60-60-50 nur maximal eine Rechenzeiteinsparung von 6 % erzielt werden konnte. Das hat nach Ansicht des Verfassers zwei Ursachen. Erstens verursacht die Lösung des Beispiels einen relativ geringen Arbeitsaufwand, wie man schon aus der vergleichsweise niedrigen Rechenzeit mit der normalen Simplex-Methode ersehen kann. Da das Dekompositionsverfahren aber einen nahezu fixen und recht hohen Organisationsaufwand nötig macht, kann dieser Mehraufwand nur bei Beispielen, die zu ihrer Lösung einen grossen Aufwand an Arbeit erfordern, kompensiert und überkompensiert werden. Darüberhinaus hat das Testprogramm, mit dem gerechnet wurde, den Nachteil, dass es nur einstufige Dekomposition zulässt, so dass die KOP stets mit m Restriktionen gerechnet werden müssen. Das ist ceteris paribus bei einem Beispiel mit 60 Restriktionen hinderlicher als bei einem Beispiel mit nur 40 Restriktionen. Dass dieser Effekt bei genügend arbeitsaufwendigen Beispielen in den Hintergrund treten kann, beweist das Beispiel B 100-50-100, das doppelt soviele Restriktionen wie Variablen hat, sowie die Beispiele B 80-500-50 und B 100-500-10.

3.2.3. Die Rechengenauigkeit

Zur Beschreibung der Rechengenauigkeit werden zwei Fehler definiert :

(1) Der maximale, absolute Zeilenfehler F_{max} :

$$F_{max} = \max_i \left| b_i - \sum_{j=1}^{n} a_{i,j} \; x_j^* - b_i^* \right| . \qquad (42)$$

Dabei ist b_i^* der Wert der Schlupfvariablen der i-ten Zeile im Optimum des letzten KOP.

(2) Der durchschnittliche, absolute Zeilenfehler F_e :

$$F_e = \frac{\sum_{i=1}^{m} \left| b_i - \sum_{j=1}^{n} a_{i,j} \; x_j^* - b_i^* \right|}{m} . \qquad (43)$$

Zu bemerken ist dazu noch, dass das Testprogramm keinerlei besondere Vorkehrungen trifft, um die Rechengenauigkeit zu erhöhen, wie etwa das Rechnen mit "doppelter Genauigkeit" oder Reinversionen von Matrizen.

a) Testbeispiel B 60-60-50 :

Für die Test-Serie mit ausschliesslich küntlichen TP sind die Rundungsfehler in Tabelle 14 aufgeführt.

Tabelle 14 : Rundungsfehler für Testbeispiel B 60-60-50 bei Dekompositions-Läufen mit ausschliesslich künstlichen TP

Anzahl der TP	Anzahl der TP-Variablen	F_{max}	$F_{\varnothing}$
2	30	$11 \cdot 10^{-9}$	$3,4 \cdot 10^{-9}$
3	20	$11 \cdot 10^{-9}$	$3,4 \cdot 10^{-9}$
4	15	$11 \cdot 10^{-9}$	$3,3 \cdot 10^{-9}$
5	12	$11 \cdot 10^{-9}$	$3,7 \cdot 10^{-9}$
6	10	$19 \cdot 10^{-9}$	$5,3 \cdot 10^{-9}$
10	6	$17 \cdot 10^{-9}$	$5,2 \cdot 10^{-9}$

Bei der Serie von Testläufen, für die normale TP in verschiedener Anzahl und Grösse eingeteilt waren, lagen die entsprechenden Fehlerwerte weit gestreut teils über den maximalen Werten teils unter den minimalen Werten der Tabelle 14, ohne dass irgendeine Regelmässigkeit erkennbar gewesen wäre.

Mit der normalen Simplex-Methode ergaben sich Fehlerwerte von $F_{max} = 32 \cdot 10^{-9}$ und

$F_{\varnothing} = 9,6 \cdot 10^{-9}$.

Das ist z.T. mehr als das Doppelte der Fehler in Tabelle 14.

b) Testbeispiel B 40-100-50 :

Für die Test-Serie mit ausschliesslich künstlichen TP ergab sich, dass F_{max} stets zwischen $9 \cdot 10^{-9}$ und $15 \cdot 10^{-9}$ und F_e stets zwischen $3{,}1 \cdot 10^{-9}$ und $5{,}1 \cdot 10^{-9}$ lag. Eine systematische Abhängigkeit dieser Werte von der Anzahl der TP war nicht erkennbar.

Bei den mit der normalen Simplex-Methode gewonnenen Ergebnissen lag F_{max} mit $82 \cdot 10^{-9}$ um das 5- bis 9fache und $F_e = 24 \cdot 10^{-9}$ um etwa das 4- bis 7fache höher.

Die entsprechenden Fehler der Test-Serie mit den TP unterschiedlicher Anzahl und Anzahl von Restriktionen lagen ohne erkennbare Systematik weit gestreut unter jenen der normalen Simplex-Methode, jedoch über denen aus der Serie mit ausschliesslich künstlichen TP.

c) Testbeispiel 100-50-100 :

Mit der normalen Simplex-Methode ergab sich :

$$F_{max} = 34 \cdot 10^{-9} \quad \text{und}$$

$$F_e = 9{,}9 \cdot 10^{-9} \; .$$

Bei den Testläufen mit ausschliesslich künstlichen TP lagen die Fehler in folgenden Grenzen :

F_{max} zwischen $17 \cdot 10^{-9}$ bis $31 \cdot 10^{-9}$ und

F_e zwischen $5{,}7 \cdot 10^{-9}$ bis $8{,}4 \cdot 10^{-9}$.

Hier liegen die Fehler nicht so erheblich unter ihren Vergleichswerten der normalen Simplex-Methode wie bei dem Beispiel B 60-60-50 oder gar B 40-100-50. Das mag an der höheren Anzahl der Restriktionen dieses Beispiels liegen. Vermutlich wäre hier unter dem Gesichtspunkt der Rechengenauigkeit eine Dekomposition des KOP-Duals angebracht, um die Restriktionen-Anzahl zu vermindern. Da kaum ein primales KOP mehr als 20 Variablen hatte, wäre

das sicherlich lohnend, da die KOP der zweiten Dekompositionsstufe nicht mehr Restriktionen haben können, als das KOP der ersten Stufe Variablen hat.

d) Testbeispiel B 40-100-100 :

Die Lösung mit der normalen Simplex-Methode ergab folgende Fehler :

$$F_{max} = 16 \cdot 10^{-9} \text{ und}$$

$$F_{\bullet} = 9{,}1 \cdot 10^{-9} .$$

Bei den Testläufen mit ausschliesslich künstlichen TP lagen die Fehler in folgenden Grenzen :

F_{max} zwischen $4{,}8 \cdot 10^{-9}$ und $7{,}5 \cdot 10^{-9}$ und

$F_{\bullet}$ zwischen $1{,}8 \cdot 10^{-9}$ und $2{,}6 \cdot 10^{-9}$.

Bei der normalen Simplex-Methode lag der absolute, maximale Fehler also stets mehr als doppelt und der absolute, durchschnittliche Fehler mehr als dreimal so hoch.

e) Testbeispiel B 80-500-50 :

Mit der normalen Simplex-Methode ergab sich :

$$F_{max} = 447 \cdot 10^{-9} \text{ und}$$

$$F_{\bullet} = 123 \cdot 10^{-9} .$$

Von dem beiden Dekompositionsläufen war einer etwas verunglückt, weil er infolge einer zu klein gewählten Prüfnull zum Schluss noch einen nicht notwendigen Zyklus gerechnet und sich dadurch wieder ein wenig von Optimum entfernt hat. Dennoch blieben auch bei diesem Lauf die Fehler unter den entsprechenden bei der normalen Simplex-Methode. Es ergaben sich nämlich :

$$F_{max} = 298 \cdot 10^{-9} \text{ und}$$

$$F_{\bullet} = 109 \cdot 10^{-9} .$$

Der zweite Dekompositionslauf verlief normal und ergab :

$$F_{max} = 149 \cdot 10^{-9} \text{ und}$$

$$F_{e} = 42 \cdot 10^{-9} .$$

Bezogen hierauf liegen die Fehler bei der einfachen Simplex-Methode um das 3fache höher.

f) Testbeispiel B 100-500-10 :

Für dieses Testbeispiel liegen nur die Fehler für die beiden Dekompositionsläufe vor, die sich für beide Läufe gleichhoch ergaben mit :

$$F_{max} = 45 \cdot 10^{-9} \text{ und}$$

$$F_{e} = 3{,}9 \cdot 10^{-9} .$$

Die hier ausgewiesenen Fehler fallen z.T. deshalb besonders niedrig aus, weil bei der Multiplikation fehlerhafter Lösungswerte mit einer Matrix, die zu 90 % Nullen enthält, die Fehler ceteris paribus weniger auf die nach der Multiplikation zu bildenden Zeilensummen durchschlagen als bei einer Matrix ohne Nullen. Das ist aus der Fehlerberechnung dieses Beispiels ersichtlich. Denn für die ersten 10 Zeilen der Matrix, die nur zu 50 % mit Nullen besetzt sind, ist der ausgewiesene Zeilenfehler in den meisten Fällen um ein Vielfaches grösser als in den letzten 90 Zeilen, die zu fast 95 % mit Nullen besetzt sind.

In der folgenden Tabelle 15 wird ein abschliessender Überblick über die bei den verschiedenen Rechnungen ausgewiesenen Fehler gegeben.

Tabelle 15 : Übersicht über die ausgewiesenen Fehler.
Alle Fehlerangaben sind mit 10^{-9} zu multiplizieren.

(1) Beispiel	F_{max}			$F_{\bullet}$		
	(2) normale Simplex-Methode	(3) Dekompositions-verfahren	(4) (4)= (2)/(3)	(5) normale Simplex-Methode	(6) Dekompositions-verfahren	(7) (7)= (5)/(6)
60- 60- 50	32	19	1,7	9,6	5,3	1,8
40-100- 50	82	15	5,5	24	5,1	4,7
100- 50-100	34	31	1,1	9,9	8,4	1,2
40-100-100	16	7,5	2,1	9,1	2,6	3,5
80-500- 50	447	135	3,3	123	43	2,9
100-500- 10	-	45	-	-	3,9	-

(3) und (6) : Diese Spalten enthalten das schlechteste Ergebnis eines Laufes aus einer Serie mit ausschliesslich künstlichen TP. Einzige Ausnahme ist das Beispiel B 80-500-50, bei dem der Lauf mit einem überflüssigen Zyklus in diesem Zusammenhang nicht gewertet wurde.

Die Spalten (4) und (7) zeigen, um das Wievielfache die bei der normalen Simplex-Methode ausgewiesenen Fehler grösser sind, als die entsprechenden Fehler des schlechtesten Dekompositionslaufes. Die Tabelle zeigt, dass das Dekompositions-Test-Programm in den Fällen nur zu einer geringen Verbesserung der Genauigkeit führt, in denen gilt $n \leq m$.

Aber einmal ist diese Bedingung bei praktischen Problemen fast nie gegeben und zum anderen ist das nur ein Mangel des Testprogramms, welches keine Dekomposition des KOP ermöglicht, und kein Mangel des Dekompositionsverfahrens.

3.2.4. Die maximale Matrixgrösse

Es soll in diesem Abschnitt noch gezeigt werden, wie gross bei den einzelnen Beispielen für jeden Rechenlauf mit nur künstlichen TP das grösste im Verlauf der Rechnung zu berechnende KOP war. Da jedes KOP genau m Restriktionen hat, ist in Tabelle 16 nur die Anzahl der KOP-Variablen des jeweils grössten KOP eines Rechenlaufes angegeben. An einigen Testbeispielen kann man verfolgen, wie diese Anzahl von der Zahl der künstlichen TP abhängt. Nur für Rechenläufe mit ausschliesslich künstlichen TP werden diese Angaben hier gezeigt. Für Rechenläufe mit extern eingeteilten TP fallen diese Werte ceteris paribus i.a. um 10 % bis 50 % höher aus.

Anhand der Tabelle 16 sieht man, dass das maximale KOP nie weniger als r Variablen hatte. Dabei ist r die Anzahl der eigentlichen primalen Variablen, die sich in der optimalen UP-Lösung befinden. Die Anzahl der KOP-Variablen geht nur einmal über r+Z hinaus, sonst bleibt sie oft noch erheblich darunter. Wenn man bei praktischen Beispielen r einigermassen abschätzen kann, hat man damit einen guten Anhaltspunkt für die zu erwartende maximale Anzahl von KOP-Variablen. Sie liegt i.a. zwischen r und r+Z. Daher kann man wohl vermuten, dass die Vorteilhaftigkeit dieses (einstufigen) Ver-

Tabelle 16 : Maximale Anzahl von KOP-Variablen verschiedener Rechenläufe mit ausschliesslich künstlichen TP

Anzahl TP (Z)	Testbeispiel Nr. B					
	60-60-50	40-100-50	100-50-100	40-100-100	80-500-100	100-500-10
	r=19	r=21	r=12	r=12	r=42	r=61
2	20	-	14	13	-	-
3	20	22	16	14	-	-
4	22	23	-	14	-	-
5	-	22	15	-	-	-
6	23	26	-	15	-	-
7	-	24	14	-	-	-
8	-	25	-	14	-	-
10	28	24	17	16	-	-
13	-	23	19	21	-	-
15	-	23	-	-	-	-
17	-	23	23	-	-	-
20	-	26	-	20	49	-
25	-	29	25	31	50	66
34	-	34	-	34	-	-
42	-	-	-	-	-	76
50	-	50	-	50	-	-

fahrens ceteris paribus um so grösser ist, je grösser die Differenz n-r ist.

Wenn man das KOP jeweils in internen Speichern einer Rechenanlage unterbringen will, so kann man aus diesen Angaben Rückschlüsse ziehen bezüglich des Bedarfs an internen Speichermöglichkeiten für Daten bei einstufiger oder zweistufiger Dekomposition.

3.2.5. Die Konvergenz des Zielfunktionswertes zum Optimalwert.

Hinsichtlich der Konvergenz des Zielfunktionswertes zum Optimalwert unterscheidet sich das Verfahren wohl kaum von anderen LP-Algorithmen. Die Annäherung an den optimalen Wert der Zielfunktion wird von Zyklus zu Zyklus tendenziell sehr viel geringer. Tabelle 17 demonstriert anhand einiger Beispiele, zu welchem Prozentsatz der optimale Wert der Zielfunktion nach jedem Zyklus realisiert wurde.

Tabelle 18 : Stand der Annäherung an den optimalen Wert der Zielfunktion nach jedem Zyklus in relativen Zahlen (Optimalwert = 100); alle Beispiele mit 25 (ausschliesslich) künstlichen TP

Zyklus Nr.	Testbeispiel Nr.				
	40-100-50	40-100-100-	100-50-100	80-500-50	100-500-10
1	93,5	111,0	107,1	77,3	48,5
2	98,5	101,0	101,1	90,0	64,0
3	99,6	100,5	100,8	95,5	80,7
4	100,0	100,1	100,0	98,2	92,0

Fortsetzung Tabelle 18

Zyklus Nr.	Testbeispiel Nr.				
	40-100-50	40-100-100	100-50-100	80-500-50	100-500-10
5	-	100,1	-	99,6	97,6
6	-	100,0	-	99,8	99,2
7	-	-	-	99,98	99,9
8	-	-	-	99,99	99,96
9	-	-	-	100,0	99,98
10	-	-	-	-	100,0

In vielen Fällen wird die letzte Hälfte der Anzahl der Zyklen dazu benötigt, die Annäherung um den letzten Prozentpunkt an den optimalen Zielfunktionswert herauszurechnen.

Dennoch kann man die Schrittweite der Annäherung an das Optimum bei praktischen Rechnungen nicht direkt als brauchbares Abbruchkriterium für die Rechnung benutzen, da sich gelegentlich auch im weit suboptimalen Zustand der aktuellen Lösung der Zielfunktionswert über mehrere Zyklen nicht ändert.

Ganz anders verhält sich der maximale Wert der aktuellen Zielfunktion des UP nach jedem Zyklus. Er wird von Zyklus zu Zyklus tendenziell zunächst zögernd und dann progressiv kleiner und kann sich zwischendurch auch mal wieder etwas erhöhen. Im letzten Zyklus ist er bei den berechneten Beispielen stets auf weniger als 1/10 000 seines Wertes im vorhergehenden Zyklus gesunken. Da diese Grösse sich so abrupt ihrem optimalen Endpunkt nähert, kann sie ebenfalls

schlecht als Kriterium für den vorzeitigen Abbruch einer Rechnung im optimumnahen Zustand benutzt werden.

Will man die Rechnung in der Nähe des Optimums abbrechen, um Rechenzeit zu sparen, so könnte man vielleicht folgendes Abbruchkriterium anwenden : Man breche die Rechnung ab, falls sich über x Zyklen der Zielfunktionswert stets verändert hat (und somit ein Abbruch infolge Degeneration ausgeschlossen ist) und der gleitende Durchschnitt der Veränderungen des Zielfunktionswertes für die letzten x Zyklen einen bestimmten Wert y unterschreitet. Für x könnte man 3 bis 4 setzen und für y 1 oder 1/2 Prozent des erwarteten optimalen Zielfunktionswertes.

4. Schlussbemerkungen

Der hier beschriebene Dekompositions-Algorithmus ist dem einfachen Simplex-Verfahren in den meisten Fällen in jeder Hinsicht überlegen. Eine Ausnahme stellen mit Sicherheit jene LP-Probleme dar, für die $m = n = r$ gilt. Wenn diese Bedingung annähernd erfüllt ist, wie etwa bei einem Gleichungssystem mit fast quadratischer Koeffizientmatrix, dann wäre das maximale, zu berechnende KOP etwa so gross wie das UP und die Anwendung dieses Verfahrens könnte gegenüber der einfachen Simplex-Methode nur Nachteile bringen.

Soll dieses Verfahren mit Vorteil anwendbar sein, so muss von den folgenden zwei Bedingungen zumindest eine erfüllt sein :

$$n > r \quad \text{und}$$

$$m > r .$$

Ist lediglich die erste Bedingung erfüllt, so kann man durch Dekomposition der ersten Stufe Variablen aussondern, die im KOP nicht mitberechnet werden.

Ist lediglich die zweite Bedingung erfüllt, so sollte man

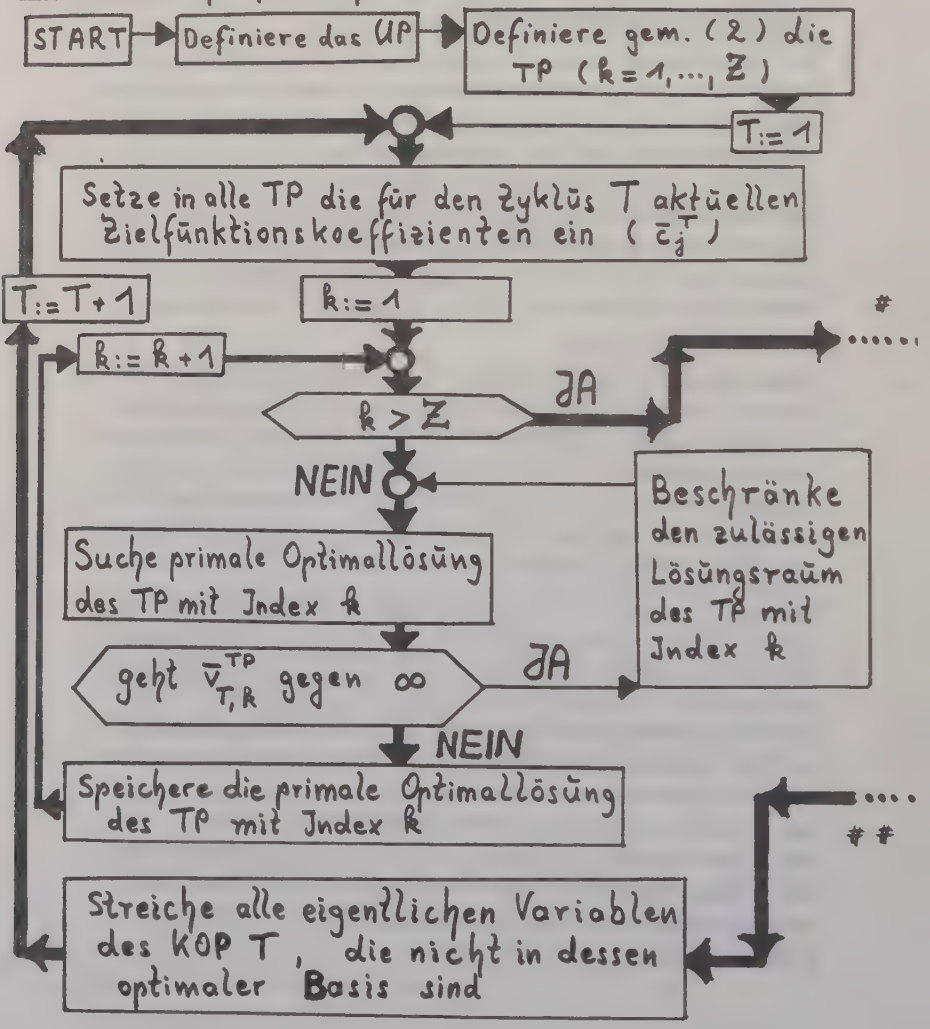
auf eine Dekomposition der ersten Stufe verzichten und direkt das UP wie ein KOP in der zweiten Stufe dekomponieren. Sind beide Bedingungen erfüllt, so kann man von beiden Dekompositionsstufen Gebrauch machen.

Wie gross dieser Unterschied zwischen n und r oder m und r sein sollte, ist vermutlich von der erforderlichen Menge an Arbeit abhängig, die z.B. die einfache Simplex-Methode für die Lösung eines LP-Problems aufwenden müsste, und von den speziellen, numerischen Schwierigkeiten, die bei der Lösung des Problems auftauchen. Bei gegebenen Relationen zwischen (m und r) und (n und r) dürfte sich die Anwendung des Dekompositionsverfahrens um so eher lohnen, je grösser die erforderlich Arbeitsmenge ist und je grösser die numerischen Schwierigkeiten sind.

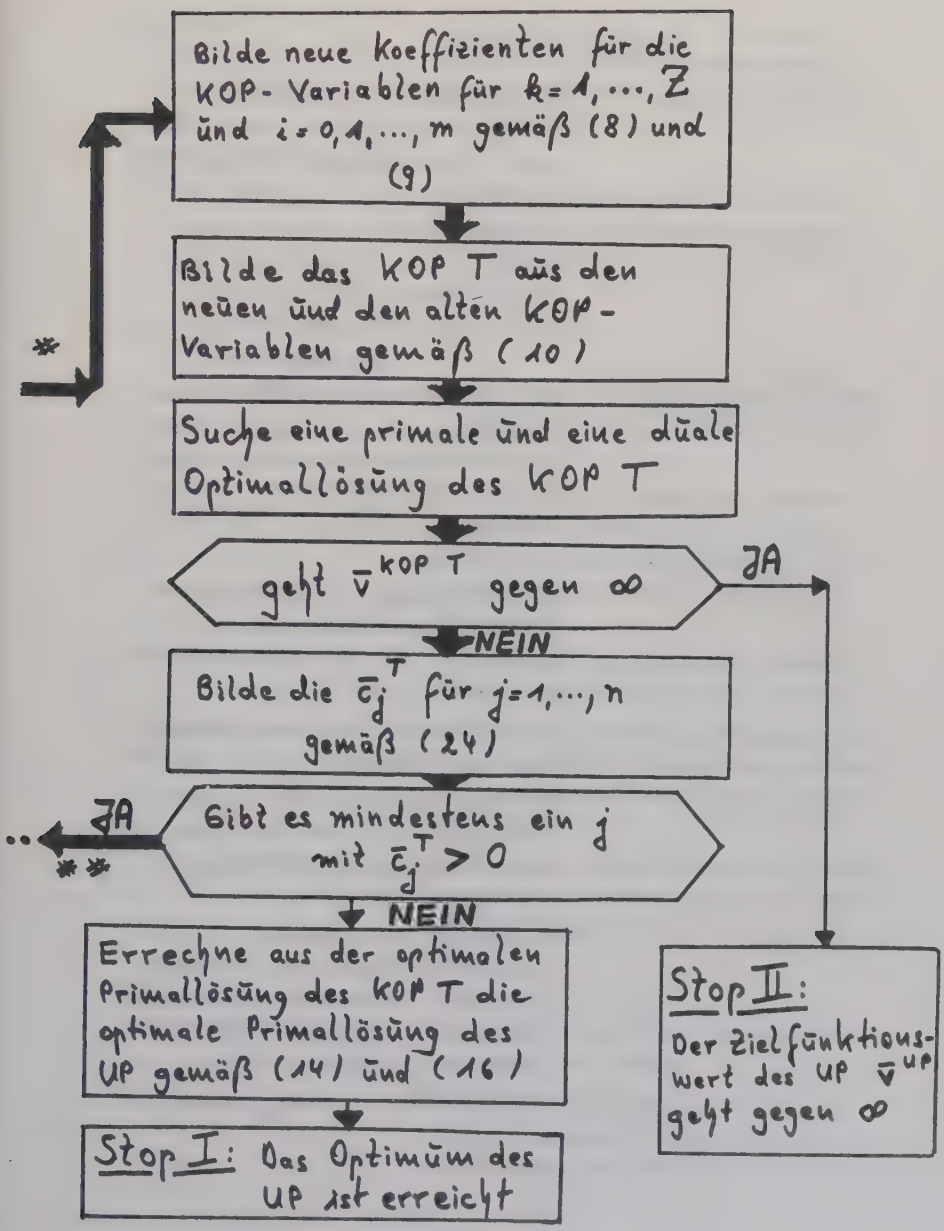
Dieses Dekompositionsverfahren zieht zwar auch einen Vorteil aus der Tatsache, dass die anfängliche Koeffizientenmatrix mit vielen Nullen besetzt ist. Es ist in seiner Vorteilhaftigkeit aber nicht wie die revidierte Simplex-Methode davon abhängig. Ausserdem kann das Dekompositionsverfahren im Gegensatz zur revidierten Simplex-Methode auch bei Problemen vorteilhaft sein, für die gilt : m = n.

Ein weiterer Punkt sollte noch erwähnt werden. Aus der Tatsache, dass bei den berechneten Beispielen durch die externe Einteilung grösserer (nicht-künstlicher) TP nur selten, und nur geringe Vorteile erzielt werden konnten, darf man noch nicht schliessen, dass sich das bei allen LP-Problemen so verhält. Denn praktisch relevante Probleme weisen in ihrer Koeffizientenmatrix meistens irgendeine Struktur auf, die man sich durch geeignete TP-Einteilung möglicherweise zunutze machen kann. Ob die mit der Einteilung grösserer TP verbundene Mehrarbeit wieder überkompensiert werden kann durch eine Verringerung der Zyklenzahl kann nur an sehr grossen Problemen getestet werden.

Flußdiagramm für ein allgemeines Dekompositionsverfahren für LP-Probleme



Z : Anzahl der TP


Bilde neue Koeffizienten für die KOP-Variablen für k = 1, ..., Z und i = 0, 1, ..., m gemäß (8) und (9)
Bilde das KOP T aus den neuen und den alten KOP-Variablen gemäß (10)
Suche eine primale und eine duale Optimallösung des KOP T
geht $\bar{v}^{KOP\ T}$ gegen ∞
JA
NEIN
Bilde die $\bar{c}_j^T$ für j = 1, ..., n gemäß (24)
Gibt es mindestens ein j mit $\bar{c}_j^T > 0$
JA
NEIN
Errechne aus der optimalen Primallösung des KOP T die optimale Primallösung des UP gemäß (14) und (16)
Stop I: Das Optimum des UP ist erreicht
Stop II: Der Zielfunktionswert des UP $\bar{v}^{UP}$ geht gegen ∞

Literatur-Verzeichnis

[1] Abadie, J.M. : Le principe de décompisition de Dantzig et Wolfe, in : Revue Française de Recherche Operationelle, 4 (1960), S. 93-115.

[2] ders. : On decomposition principle, in : Bericht Nr. 63-20 des Operations Research Center der Universität Californien, Berkeley, 1963.

[3] Abadie, J.M., u. A.C. Williams : Dual and parametric methods in decomposition, in : Recent advances in mathematical programming, Hrsg. : Graves, R., u. P. Wolfe, Mc Graw-Hill (1963), S. 149-158.

[4] Adam, D., u. W. Röhrs : Ein Algorithmus zur Dekomposition linearer Planungsprobleme, in : Zeitschrift für Betriebswirtschaft, 37 (1967), S. 395-417.

[5] Adam, D. : Entscheidungsorientierte Kostenbewertung, Wiesbaden 1970, S. 196-230.

[6] Balas, E. : An infeasibility-pricing decomposition method for linear programs, in : Operations Research (ORSA), 14 (1966), S. 847-873.

[7] ders. : Une methode de decomposition quasi-primale-duale pour des programmes lineaires, I-Variante, in : Comptes Rendus de l'Academie des Sciences Paris, 261 (1965), 4. Oktober, S. 2572-2574.

[8] ders. : idem, II-Variante, in : Comptes Rendus de l'Academie des Sciences Paris, 261 (1965), 18. Oktober, S. 2809-2811,

[9] Baumol, W.J., u. T. Fabian : Decomposition, pricing for decentralisation and external economies, in : Management Science, 11 (1964), S. 1-32.

[10] Beale, E.M.L., P.A.B. Hughes u. R.E. Small : Experiences in using a decomposition program, in : The Computer, 8 (1965), S. 13-18.

[11] Bell, E.J. : Primal-dual decomposition programming, in : Bericht Nr. 65-23 des Operations Research Center der Universität Californien, Berkeley, August 1965.

[12] Benders, J.F. : Partition in mathematical programming, Dissertation, Utrecht (1960).

[13] ders. : Partitioning procedures for solving mixed-variables programming problems, in : Numerische Mathematik, 4 (1962), S. 238-252.

[14] Bennet, J.M. : An approach to some structured linear programming problems, in : Operations Research (ORSA), 14 (1966), S, 636-645.

[15] Bennet, J.M. u. D.R. Greene : idem, in : Operations Research (ORSA) 17 (1969), S. 749-750.

[16] Dantzig, G.B. : Lineare Programmierung und Erweiterungen, Springer-Verlag, Berlin, Heidelberg, New York 1966.

[17] Dantzig, G.B., u. M.B. Shapiro : Solving the chemical equilibrium problem using the decomposition principle, in : RAND-Report P-2056, 10. August 1960.

[18] Dantzig, G.B., u. P. Wolfe : A decomposition principle for linear programs, in : RAND-Report P-1544, 10. November 1958, revidiert : 6. November 1959.

[19] dies. : idem, Operations Research, 8 (1960), S. 101-111.

[20] dies. : The decomposition algorithm for linear programs, in : Econometrica, 29 (1961), S. 767-778.

[21] Grosse, L. : Lösung grösserer linearer Optimierungsaufgaben mit dem Dekompositionsverfahren nach Dantzig-Wolfe, in : Wissenschaftliche Zeitschrift der Hochschule für Ökonomie Berlin, 11 (1966), Heft 2, S. 159-161.

[22] ders. : idem, in : H. Bader u.a. (Hrsg.) : Mathematik und Wirtschaft, Band 4, Westdeutscher Verlag, Köln, Opladen, 1967.

[23] Harvey, R.P. : The decomposition principle for linear programming, in : International Journal of Computer Mathematics, 1 (1964), S. 20-35.

[24] Heestermann, A.R.G. : Special simplex algorithm for multisector problems, in : Numerische Mathematik, 12 (1968), S. 288-306.

[25] Heesterman, A.R.G., u. J. Sandee : Special simplex algorithm for linked problems, in : Management Science, 11 (1965), S. 420-427.

[26] Heinemann, H. : Ein allgemeines Dekompositionsverfahren für lineare Optimierungsprobleme, in : Zeitschrift für betriebswirtschaftliche Forschung, 22 (1970), S. 302-317.

[27] Hu, T.C. : A decomposition algorithm for shortest paths in an network, in : IBM Research Report RC-1562, Februar 1966.

[28] ders. : idem, in : Operations Research (ORSA), 16 (1968), S. 91-102.

[29] Koenig, J.W.J. : Dynamische Optimierungsmodelle der chemischen Industrie, Dissertation, Hamburg 1968.

[30] Künzi, H.P., u. S.T. Tan : Lineare Optimierung grosser Systeme, in : Lecture Notes in Mathematics, 27 (1966), Hrsg. : Dold, A., u. B. Eckmann, Springer-Verlag, Berlin, Heidelberg, New York, 1966.

[31] Lüttgen, H. : Reduktion und Dekomposition von PERT-Netzen, in : Ablauf- und Planungsforschung, 8 (1967), S. 370-379.

[32] ders. : Reduktion und Dekomposition von Planungsnetzen, in : Ablauf- und Planungsforschung, 7 (1966), S. 92-104.

[33] Macguire, C.B. : Some extensions of the Dantzig-Wolfe decomposition scheme, Center for Research in Management Science, Universität Californien, Berkeley, Working Paper Nr. 66, März 1963.

[34] Marschak, J. : Computation, decomposition an internal pricing, Center for Research in Management Science, Universität Californien, Berkeley, Working Paper Nr. 56, Dezember 1962.

[35] Mills, G. : A decomposition algorithm for the shortest route problem, in : Operations Research (ORSA), 14 (1966), S. 279-291.

[36] Müller-Merbach, H. : Das Prinzip der "direkten Dekomposition" in der linearen Planungsrechnung, in : Zeitschrift für Angewandte Mathematik und Mechanik, 46 (1966), Sonderheft, S. T 102-T 104.

[37] ders. : Das Verfahren der direkten Dekomposition in der linearen Planungsrechnung, in : Ablauf- und Planungsforschung, 6 (1965), S. 306-322.

[38] Ohse, D. : Numerische Erfahrungen mit zwei Dekompositionsverfahren der linearen Planungsrechnung, in : Ablauf- und Planungsforschung, 8 (1967), Heft 2, S. 289-301.

[39] Nemhauser, G.L. : Decomposition for linear programming by dynamic programming, in : Naval Research Logistics Quarterly, 11 (1964), S. 191-195.

[40] Parikh, S.C., u. W.S. Jewell : Decomposition of project network, in : Management Science, 11 (1965), S. 444-459.

[41] Rech, P. : Decomposition and interconnected systems in mathematical programming, Bericht Nr. 65-31 des Operations Research Center, Universität Californien, Berkeley, Sept. 1965.

[42] Rosen, J.B. : Convex partition programming, in : Recent advances in mathematical programming, Hrsg. : Graves, R., und P. Wolfe, Mc Graw-Hill 1963, S. 159-176.

[43] ders. : Primal partition programming for block diagonal matrices, in : Numerische Mathematik, 6 (1964), S. 250-260.

[44] Rosen, J.B., und J.C. Ornea : Solution of nonlinear programming problems by partitioning, in : Management Science, 10 (1963/1964), S. 160-173.

[45] Sanders, J.L. : A nonlinear decomposition principle, in : Operations Research (ORSA), 13 (1965), S. 266-271.

[46] Tan, S.T. : Beiträge zur Dekomposition von linearen Programmen (I und II), in : Unternehmensforschung, 10 (1966), S. 168-189 und 247-268.

[47] Tournol du Clos, J. : Note sur la decomposition matricielle triangulaire. Application a l'inversion matricielle et la resolution des systems lineaires, in : Revue Francaise d'informatique et de recherche operationelle, Nr. 3, Mai-Juni 1967.

[48] Tuy, P.H. : Sur le probleme des constraintes supplementaires en programmation lineaire et son application au probleme de decomposition, in : Elektronische Informationsverarbeitung und Kybernetik, 3 (1967) 43, S. 141-156.

[49] Whinston, A. : A decomposition algorithm for quadratic programming, Cowles Foundation, Discussion Paper 172, Yale University, Juni 1964.

[50] ders. : A dual decomposition algorithm for quadratic programming, in : Cahiers du Centre d'Etudes de Recherche Operationelle, 6 (1964), S. 188-201.

[51] Zangwill, W.I. : A decomposable nonlinear programming approach, in : Operations Research (ORSA), 15 (1967), S. 1068-1087.